Classifying multivariate time series data: An empirical comparison of similarity measures

Hassan Khotanlou[1], Amir Salarpour[2], Pedram MohajerAnsari[2]

1: RIV lab, Department of computer engineering, Bu-Ali Sina University, Hamedan, Iran

Tel: +98- 81- 38272006; fax: +98-81-38292631

2: AI Lab, Department of computer engineering, Sirjan University of Technology, Kerman, Iran

Tel: +98-34-41522045; fax: +98-34- 41522045

Hassan Khotanlou (khotanlou@basu.ac.ir)

Amir Salarpour (salarpour@sirjantech.ac.ir)

Pedram MohajerAnsari(pedram.mohajer@outlook.com)

Abstract

Multivariate time series (MTS) data are widely used in different fields, and assessing their similarity is a main task of many computational systems. While the researches in this field have been focused on proposing novel similarity measures for the underlying data, this field suffers from a lack of comparative studies using quantitative and large scale evaluations. In order to provide a comprehensive validation, the effectiveness of the 14 well-known similarity measures with on 32 MTS datasets coming from a wide variety of application domains were re-implemented and tested, and the empirical comparison regarding their effectiveness based on nearest neighbor classification task was outlined. In addition, the comparative experimental findings regarding similarity measures effectiveness was presented. Furthermore, the statistical significance tests were used to derive meaningful conclusions. The results provide a comparative background between similarity measures to find the most proper method in terms of performance this field.

Keywords- Multivariate time series, similarity measures, elastic measures, classification, evaluation.

1. Introduction

Recently, multivariate time series (MTS) data has been considered as an important resource to analyze the recorded dynamics of scientific mechanism over time (Aghabozorgi, Shirkhorshidi, & Wah, 2015; Fu, 2011; E Keogh, 2011). Examples of real-world applications include analyzing the fluctuations of the stock market (financial data analysis), interpreting the electrocardiogram data (medical data processing), weather forecasting, recognizing trajectory patterns, and identifying moving object (motion data analysis) (Eamonn Keogh, et al., 2009; Liao, 2005). There are other type of data, such as signature or object contours, that can be converted to MTS data for further analyzing (Eamonn Keogh, et al., 2009). The MTS data are often enriched with additional information such as class labels, place and occurrence time (Fu, 2011).

The MTS data mining goal is to extract the meaningful knowledge from the shape of data (Eamonn Keogh & Kasetty, 2003). A key concept toward dealing with MTS data mining is determining their pairwise similarity. In fact, the MTS similarity (or dissimilarity) measure is the core component to many mining tasks like retrieval, clustering, and classification. Furthermore, deriving a distance that correctly captures the semantics and reflects the underlying data similarity is not straightforward (Serra & Arcos, 2014). The main challenges of estimating MTS similarity are high-dimensionality of data and efficiency (Kleist, 2015).

In the recent decade, researches on time series similarity measures have attracted much attention. Many techniques have been proposed to measure the similarity of time series data. Although the literature covers a wide variety of such similarity measures, this work will focus on the most cited techniques which emerge repeatedly throughout the related works. For example, Euclidean distance (ED) (Faloutsos, Ranganathan, & Manolopoulos, 1994; Eamonn Keogh & Kasetty, 2003), Manhattan Distance (MD) (Eamonn Keogh & Kasetty, 2003), Dissimilarity Metric (DISSIM) (Frentzos, Gratsias, & Theodoridis, 2007), Dynamic Time Warping (DTW) (Berndt & Clifford, 1994; Rabiner & Juang, 1993), Longest Common Subsequence (LCSS) (Vlachos, Kollios, & Gunopulos, 2002), Edit Distance with Real Penalty (ERP) (Chen & Ng, 2004), Edit Distance on Real sequence (EDR) (Chen, Özsu, & Oria, 2005), Sequence Weighted Alignment model (SWALE) (Morse & Patel, 2007), Time Warp Edit Distance (TWED) (Marteau, 2009), the Move-Split-Merge distance(MSM) (Stefan, Athitsos, & Das, 2013), Hausdorff (Hausdorff, 1927), Fréchet (Eiter & Mannila, 1994; Fréchet, 1906), and Symmetrized Segment-Path Distance (SSPD) (Besse, Guillouet, Loubes, & Royer, 2016).

Few researchers have addressed the problem of finding the best similarity measure for time series analysis. Preliminary work was carried out by Ding et al. (Ding, Trajcevski, Scheuermann, Wang, & Keogh, 2008), who compared and discussed nine different similarity measures over 38 diverse fix-length univariate time series datasets for the classification task. They found that the performance of most of the elastic similarity measures are very close to the DTW one. Another work was done by Wang et al. (X. Wang, et al., 2013), they found the same results based on fix-length univariate time series. The most interesting experiments to this issue has been employed through the classification task by Serra and Arcos (Serra & Arcos, 2014). The results obtained by them suggested that the TWED similarity measure is consistently significantly outperform other seven competitor measures. They also assessed the best parameter choices for the similarity measures. In two recent papers by Lines and Bagnall (Lines & Bagnall, 2015) and Bagnall et al.

(Bagnall, Bostrom, Large, & Lines, 2016), several similarity measures were evaluated for ensemble classification, and they have concluded that there is no significant difference between elastic similarity measures and using the ensemble classifier can produce significantly better performer than any other elastic similarity measures. A key limitation of these researches is that all of the comparative studies have considered the fix-length univariate time series data. Even despite some works in the area, unfortunately it is still unclear which similarity measure is more appropriate for the MTS data classification. However, with the multitude of competitive techniques, we believe that there is a strong need for a comprehensive comparison of similarity measures in MTS data classification context that has drawn the most attention from data mining researchers. In every newly proposed similarity measure a kind of superiority over some of the existing methods has been claimed. On the other hand, their empirical evaluations have not been the same and perhaps adequate. This has not only confused newcomers and specialists, but also led to the use of a wrong method based on incomplete and not generalized results (Wang, Su, Zheng, Sadiq, & Zhou, 2013).

To address these problems, an empirical evaluation of similarity measures for MTS classification was performed. As for the considered measures, nine elastic similarity measures were included, as these were found the state-of-the-art similarity measures. Apart from these nine measures, the three geometry-based and two differential-based similarity measures were chosen. The main contributions of this work can be phrased as:

- Presentation of an extensive summary and background of the considered similarity measures, with basic formulations.
- Reimplementation of 14 different similarity measures in MATLAB and Mex.

- Estimation of similarity measures effectiveness and efficiency for nearest neighbor classification over 32 various MTS datasets.
- Assessment of the chosen parameters for each similarity measure and dataset.
- Usage of statistical significance tests in order to evaluate the superiority of given similarity measures.

The rest of the paper was organized as follows. Section 2 outlines the preliminaries related to the MTS analysis context. Section 3 reviews the most commonly used MTS similarity measures and outline the details of calculation. In section 4, the evaluation framework is explained. Section 5 reports the main contribution of this work – the results of the experimental evaluations of different similarity measures. In section 6 the paper is concluded and the possible future work is discussed.

2. Preliminaries

The MTS data is a finite temporal sequence that is sampled from the continuous data. For simplicity and without any loss of generality, the MTS data are considered discrete hereafter.

Definition 1. A set of MTS data is defined by:

$$X = \{A_1^p \vee p \in N\} \tag{1}$$

where A_1^p is a finite multivariate time series with discrete samples and temporal indexes from 1 to p.

Definition 2. Each MTS data is formally defined as a sequence of pairs as follows:

$$A_1^p = \{(a_1, t_1), \dots, (a_i, t_i), \dots, (a_p, t_p)\} \tag{2}$$

where $a_i \in R^d$ is the i-th sample of time series in $d \geq 1$ dimensional space, $t_i \in R$ is the time-stamp variable for a_i with the condition that if $i > j$ then $t_i > t_j$ (time stamp is strictly increase with the sequence of samples).

Definition 3. A sub time series is defined by:

$$A_i^j = \{(a_i, t_i), \dots, (a_j, t_j)\} \tag{3}$$

where A_i^j is a sub time series consisting i-th sample to j-th sample of A. If $j < i$ the A_i^j is the null time series and noted by $\emptyset$. $|A|$ denotes the length (the number of samples) of A.

Definition 4. A piecewise linear MTS is a set of line segments that are bounded between successive MTS data samples which is defined as:

$$As_1^p = (as_1, \dots, as_i, \dots, as_{p-1}) \tag{4}$$

where each $as_i \in R^{2\times d}$ is a line-segment $\overline{a_i a_{i+1}}$ of A_i^j that is bounded between a_i and a_{i+1}.

Definition 5. As a base measure to find the distance between MTS data samples, the Euclidean metric (Faloutsos, et al., 1994) is used in the rest of this paper as follows:

$$d_{eucl}(a_i, b_j) = \sqrt{\sum_{dim=1}^{d} \left(a_i^{dim} - b_j^{dim}\right)^2} \tag{5}$$

where $d_{eucl}(a_i, b_j)$ is the Euclidean distance between two d-dimensional time series samples a_i and b_j.

Definition 5. Furthermore, the point-to-segment distance (Besse, et al., 2016) is defined as a minimum Euclidean distance between sample points and given MTS segment as revealed by:

$$d_{p2s}(a_i, bs_j) = \begin{cases} d_{eucl}(a_i, a_i^{proj}) & if\, a_i^{proj} \in bs_j \\ min \begin{cases} d_{eucl}(a_i, b_j) \\ d_{eucl}(a_i, b_{j+1}) \end{cases} & otherwise \end{cases} \quad (6)$$

where a_i^{proj} is the orthogonal projection of MTS data sample a_i on the MTS segment bs_j and $d_{p2s}(a_i, bs_j)$ is the point-to-segment distance between $x_{k,i}$ and $xs_{l,j}$.

Most of similarity measures in the literature are considered only the univariate time series. There exist two ways to generalize the similarity measures to multi-dimensional case: the first one is calculating similarity measure for each dimension independently and then sum up distances of all dimensions, the second was is to estimate the similarity for all dimensions identically and together, calculate the distance between MTS samples in multidimensional space (Shokoohi-Yekta, Wang, & Keogh, 2015). In this paper it is assumed that all dimensions of MTSs are synchronized and relevant, so the dependently method to generalization from one-dimensional to multi-dimensional time series is used.

3. Multivariate time series similarity measures

In this section, the main time series similarity measures developed in the literature are reviewed. The MTS similarity measures compare the overall shape of the MTS by measuring the closeness of time series. The MTS similarity measures can be divided into four main categories as follows:

3.1. Lock-step measures

The measures in this category compare the MTS samples that have the exact same temporal index. This kind of measures are limited to MTS with equal length. In the cases that the time series have not the same length, the re-sampling can be used. Figure 1 shows the intuition behind Lock-step measures.

Euclidean Distance (ED)

Measures based on L_p norms are the members of lock-step measures (Faloutsos, et al., 1994). Generally the ED is L_2 norm which is one of the typical ways to estimate the dissimilarity between time series samples.

$$D_{eucl}\left(A_1^p, B_1^q\right) = \frac{1}{p}\sum_{i=1}^{p} d_{eucl}(a_i| \quad |, b_i) \qquad (7)$$

where $p = q$ is the length of A that is equal to the length of B, and a_i and b_i are the i-th sample of A and B time series, respectively.

There are some other lock-step measures (e.g. Manhattan distance (Eamonn Keogh & Kasetty, 2003), DISSIM (Frentzos, et al., 2007), or correlation (Wei, 1994)) to compare the MTS data, that is not considered as a case because of requiring the equal length MTS.

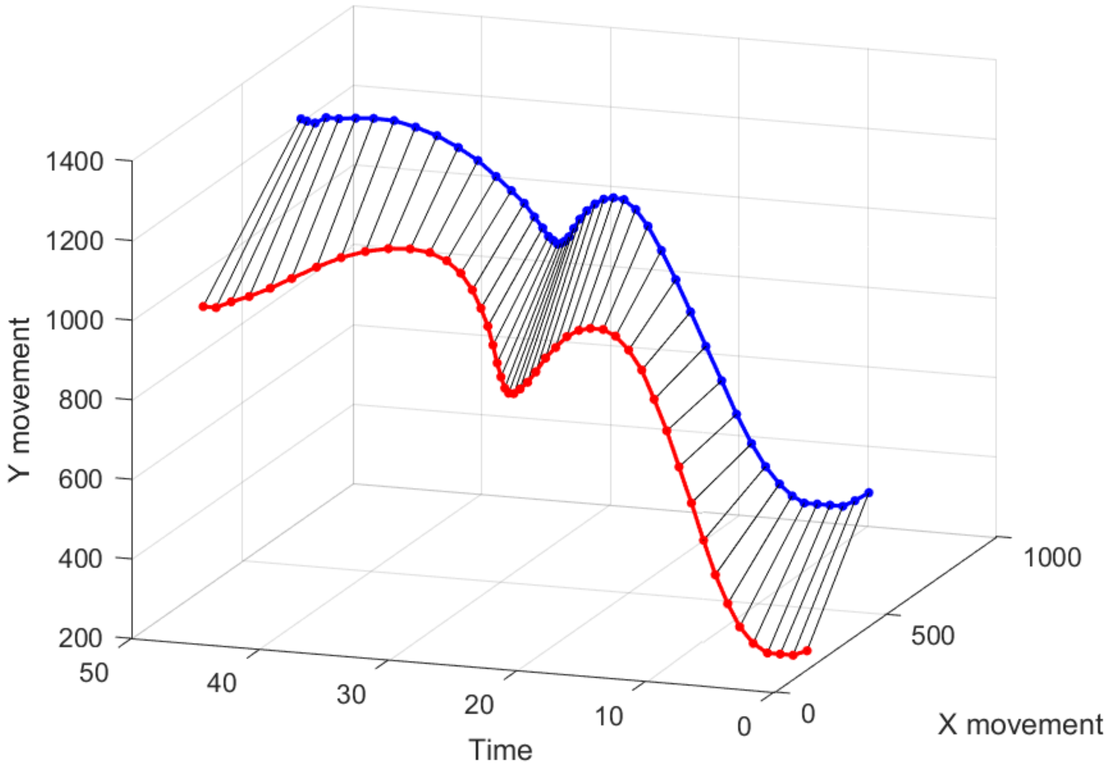


Fig. 1 An illustration of a Lock-step measure (one-to-one mapping of MTS samples).

3.2. Elastic measures

In the elastic measures category, the problem of aligning MTS with different speeds, different sampling rates, or inconsistent temporal scales is resolved by warping the temporal dimension. The basic idea of these methods is the Levenstein Distance (LD) (Levenshtein, 1966), also known as edit distance , that is the smallest number of insertion, deletion, and substitution needed to change one string to another. The edit distance between any two MTS A and B with a finite length could be defined as:

$$D\left(A_1^p, B_1^q\right) = min\begin{cases} D\left(A_1^{p-1}, B_1^q\right) + cost_{delete}\left(a_p\right) & delete \\ D\left(A_1^{p-1}, B_1^{q-1}\right) + cost_{match}\left(a_p, b_q\right) & match \\ D\left(A_1^p, B_1^{q-1}\right) + cost_{insert}\left(b_q\right) & insert \end{cases} \qquad (8)$$

where $p \geq 1$, $q \geq 1$, and $cost$ is a function that return a non-negative real number for each edit operation. Wagner and Fisher (Wagner & Fischer, 1974) developed an algorithm to calculate the LD in quadratic time using Dynamic Programming (DP) (Bellman, 1956).

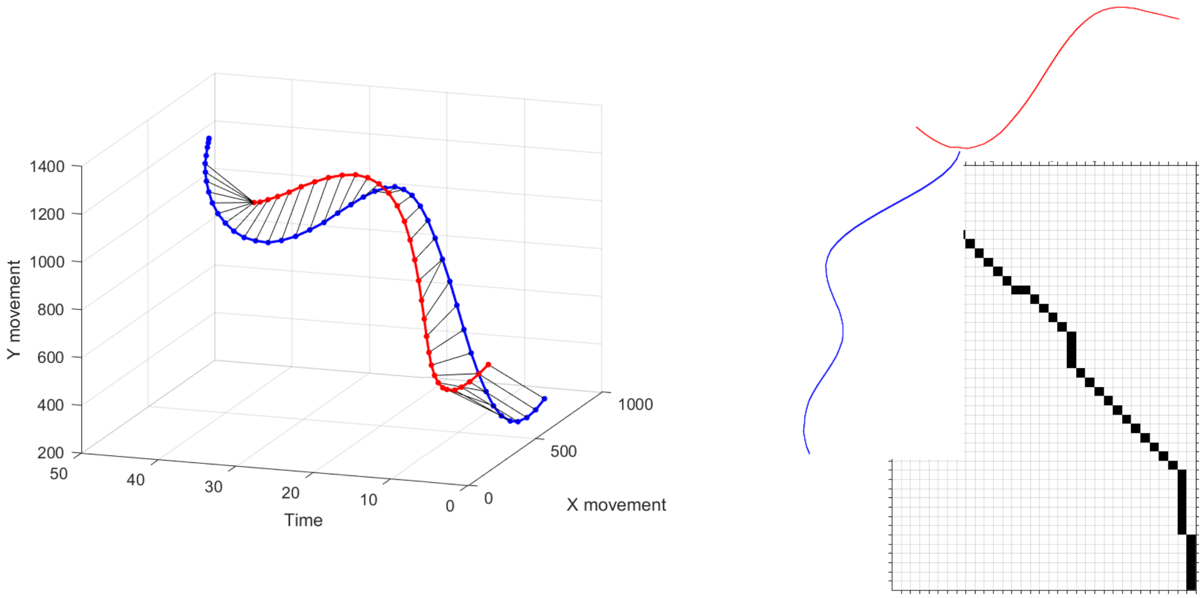
They construct a p-by-q matrix where the $\left(i^{th}, j^{th}\right)$ element of matrix contains the associated edit cost between a_i and b_j. A warping path, $W = w_1, w_2, \ldots, w_k, \ldots, w_K$, is a sequence of matrix elements that defines a mapping between A_1^p and B_1^q. The warping path is typically contingent on the following constraints:

Boundary conditions: $w_1 = D(1,1)$ and $w_K = D(p,q)$. This restricts the warping path to start in (1,1) cell of the matrix and finishes in (p,q) cell of matrix.

Continuity: given $w_k = D(i_k, j_k)$, then $w_{k-1} = D(i_{k-1}, j_{k-1})$, where $i_k - i_{k-1} \leq 1$ and $j_k - j_{k-1} \leq 1$. This forces the consecutive elements of path, W, to be the adjacent matrix cells (including diagonally adjacent cells).

Monotonicity: given $w_k = D(i_k, j_k)$, then $w_{k-1} = D(i_{k-1}, j_{k-1})$, where $i_k - i_{k-1} \geq 0$ and $bj_k - j_{k-1} \geq 0$. This limits the elements of the warping path to be monotonic in the temporal dimension.

There are many warping paths that satisfy the above conditions. However, the path that minimizes the warping cost gives us the value of LD. The elastic distance and warping path between two 2-d MTS sample with different lengths is given in Fig. 2.



(a) (b)

Fig. 2 A sample of MTS alignment. (a) two MTS with elastic distance. (b) corresponding warping path.

3.2.1. DTW

The DTW (Berndt & Clifford, 1994; Rabiner & Juang, 1993), which shares many similarities with LD, was proposed to align MTS with time shift tolerances as follows:

$$D_{DTW}\left(A_1^p, B_1^q\right) = d_{eucl}\left(a_p, b_q\right) + \min \begin{cases} D_{DTW}\left(A_1^{p-1}, B_1^q\right) \\ D_{DTW}\left(A_1^{p-1}, B_1^{q-1}\right) \\ D_{DTW}\left(A_1^p, B_1^{q-1}\right) \end{cases} \tag{9}$$

DTW distance applies local scaling of the temporal dimension that guarantees to keep the order of MTS samples and is sensitive to noise. One of the main restrictions of DTW is that it does not comply with the triangle inequality (non-metric) (Eamonn Keogh & Kasetty, 2003).

3.2.2. Constrained DTW

Constrained DTW (cDTW) (Sakoe & Chiba, 1978) is one of the most useful variants of DTW to speed up and control the deviation from the diagonal path (one-to-one matching). The cDTW similarity measure constrains the temporal scaling with Sakoe-Chiba Band, which consider a sliding window for temporal deviation and calculated as follows:

$$D_{cDTW}\left(A_1^p, B_1^q\right) = \begin{cases} d_{eucl}\left(a_p, b_q\right) + \min \begin{cases} D_{cDTW}\left(A_1^{p-1}, B_1^q\right) \\ D_{cDTW}\left(A_1^{p-1}, B_1^{q-1}\right) \\ D_{cDTW}\left(A_1^p, B_1^{q-1}\right) \end{cases} & \text{if} |p-q| \le \delta \\ \infty & \text{otherwise} \end{cases} \tag{10}$$

where δ is a windows size parameter to control the temporal deviation. The size of sliding windows greatly affects the quality of calculated similarity measure.

3.2.3. Weighted DTW

Weighted DTW (WDTW) (Jeong, Jeong, & Omitaomu, 2011) technique weighs each MTS sample according to the temporal deviation. Actually, it is a multiplicative penalty weight function that is based-on the warping difference. It is try to reduce the warping and it is a soft version of cut-off cDTW.

$$D_{WDTW}\left(A_1^p, B_1^q\right) = w\left(a_p, b_q\right) + \min \begin{cases} D_{WDTW}\left(A_1^{p-1}, B_1^q\right) \\ D_{WDTW}\left(A_1^{p-1}, B_1^{q-1}\right) \\ D_{WDTW}\left(A_1^p, B_1^{q-1}\right) \end{cases} \tag{11}$$

A logistic weight function is used as follows:

$$w\left(a_p, b_q\right) = \frac{w_{max}}{1+e^{-g\times(|p-q|-m/2)}} \times d_{eucl}\left(a_p, b_q\right) \tag{12}$$

where the w_{max} is the upper bound (set to 1), m is the average of given MTS lengths, and g is the parameter to control the penalty for large warping (the greater g, the larger penalty). The WDTW method is trying to maximize the effectiveness with optimizing the g value based on the different applications.

3.2.4. LCSS

LCSS (Vlachos, et al., 2002) measure is the longest common subsequence between MTS based on the concept of edit distance. The original LCSS measure is increased with the matching concept between two sequences. LCSS similarity measure is robust against noise by using the threshold value on the distances between compared time series samples. The LCSS method was proposed to address the robustness against outliers in time series matching.

$$LCSS\left(A_1^p, B_1^q\right) = \begin{cases} LCSS\left(A_1^{p-1}, B_1^{q-1}\right) + 1 & d_{eucl}\left(a_p, b_q\right) < \epsilon \\ \max \begin{cases} LCSS\left(A_1^{p-1}, B_1^q\right) \\ LCSS\left(A_1^p, B_1^{q-1}\right) \end{cases} & \text{otherwise} \end{cases} \tag{13}$$

where ϵ is the threshold for distance between MTS samples.

The LCSS measure is originally a similarity measure to transform it to the distance (dissimilarity measure) the following formula is used.

$$D_{LCSS}\left(A_1^p, B_1^q\right) = 1 - \frac{LCSS\left(A_1^p, B_1^q\right)}{min\,\{p,q\}} \tag{14}$$

where p and q are the length of given MTSs.

3.2.5. EDR

EDR (Chen & Ng, 2004) is another edit distance based similarity measure that works by assigning the quantified value, set to 0 or 1, to the distance between MTS samples as follows:

$$D_{EDR}\left(A_1^p, B_1^q\right) = \min \begin{cases} D_{EDR}\left(A_1^{p-1}, B_1^{q-1}\right) + cost_{EDR}\left(a_p, b_q\right) \\ D_{EDR}\left(A_1^{p-1}, B_1^q\right) + 1 \\ D_{EDR}\left(A_1^p, B_1^{q-1}\right) + 1 \end{cases} \tag{15}$$

EDR measure is also used the threshold parameter, called ϵ, to control the noise in matched MTS samples.

$$cost_{EDR}\left(a_p, b_q\right) = \begin{cases} 0 & \text{if} d_{eucl}\left(a_p, b_q\right) < \epsilon \\ 1 & \text{otherwise} \end{cases} \tag{16}$$

The EDR similarity measure attempt to remove the noise effects by quantizing the distance to two values, 0 or 1, and improve the accuracy by assigning the penalty to the unmatched time series samples.

3.2.6. ERP

ERP (Chen, et al., 2005) is an edit-based measure that uses the merits of DTW and EDR, by considering a reference point for computing the distance where there is a gap in MTS aligning. The motivation for introducing the ERP is making EDR to a metric distance with a real penalty that defined by the distance to the reference point.

$$D_{ERP}\left(A_1^p, B_1^q\right) = \min \begin{cases} D_{ERP}\left(A_1^{p-1}, B_1^q\right) + d_{eucl}\left(a_p, gap\right) \\ D_{ERP}\left(A_1^{p-1}, B_1^{q-1}\right) + d_{eucl}\left(a_p, b_q\right) \\ D_{ERP}\left(A_1^p, B_1^{q-1}\right) + d_{eucl}\left(gap, b_q\right) \end{cases} \tag{17}$$

where gap is a d-dimensional reference point to calculate the penalty for gap in MTS matching.

3.2.7. SWALE

Morse and Patel (Morse & Patel, 2007) proposed SWALE similarity measure based on edit distance that combines match rewards and gap penalties. In addition, the matching threshold is still used to find matching against noise.

$$D_{SWALE}\left(A_1^p, B_1^q\right) = \begin{cases} D_{SWALE}\left(A_1^{p-1}, B_1^{q-1}\right) + r_m & \text{if} d_{eucl}\left(a_p, b_q\right) < \epsilon \\ \max\begin{cases} D_{SWALE}\left(A_1^{p-1}, B_1^{q}\right) + g_c \\ D_{SWALE}\left(A_1^{p}, B_1^{q-1}\right) + g_c \end{cases} & \text{otherwise} \end{cases} \tag{18}$$

where ϵ is a distance threshold, r_m is a reward parameter for matching samples, and g_c is the penalty for gap in MTS alignment. The SWALE measure make an effort to achieve the distance function that could be tuned by the domain expert, with the particular knowledge, for optimal performance instead of using the same distance function for all time series domains.

3.2.8. TWED

Marteau et al. (Marteau, 2009) introduced TWED metric distance that encompasses both LCSS and DTW characteristics. They redefined edit distance operations for measuring the similarity. The originality of TWED lies in the way to control the stiffness, which is a multiplicative penalty that penalizes the deviation in the temporal dimension in a way like WDTW, unlike the constrained DTW that limits the deviation in the temporal dimension. The penalty value, namely λ is applied when MTS samples do not match.

$$D_{TWED}\left(A_1^p, B_1^q\right) = min\begin{cases} D_{TWED}\left(A_1^{p-1}, B_1^{q-1}\right) + cost_{match}\left(a_p, b_q\right) \\ D_{TWED}\left(A_1^{p-1}, B_1^{q}\right) + cost_{delete}\left(a_p, a_{p-1}\right) \\ D_{TWED}\left(A_1^{p}, B_1^{q-1}\right) + cost_{delete}\left(b_q, b_{q-1}\right) \end{cases} \tag{19}$$

$$cost_{match}(a_p, b_q) = 2 * \nu * |p - q| + d_{eucl}(a_p, b_q) + d_{eucl}(a_{p-1}, b_{q-1}) \tag{20}$$

$$cost_{delete}(a_i, b_j) = \nu + \lambda + d_{eucl}(a_i, b_j) \tag{21}$$

where ν is the stiffness parameter that control the temporal deviation, and λ parameter that penalize the gap in MTS matching.

3.2.9. MSM

Stefan et al. (Stefan, et al., 2013) introduced the MSM metric that conceptually is an edit-based approach. In this method, the similarity between MTSs is estimated by presenting a set of new operations. Move, split and merge defined as three MSM operations with an associated cost. The Move operation is equivalent to substitute operation in the edit-based distance. Split and merge operations are different from insertion and deletion, however, it is achievable by combining MSM operations. The operation cost is not the same and they depend on the value of adjacent MTS samples.

$$D_{MSM}(A_1^p, B_1^q) = \min \begin{cases} D_{MSM}(A_1^{p-1}, B_1^{q-1}) + d_{eucl}(a_p, b_q) \\ D_{MSM}(A_1^{p-1}, B_1^{q}) + cost_{MSM}(a_p, a_{p-1}, b_q) \\ D_{MSM}(A_1^{p}, B_1^{q-1}) + cost_{MSM}(b_q, a_p, b_{q-1}) \end{cases} \tag{22}$$

$$cost_{MSM}(a_1, a_2, a_3, c) = \begin{cases} c & \text{if} a_2 \le a_1 \le a_3 \text{or} a_2 \ge a_1 \ge a_3 \\ c + \min \begin{cases} d_{eucl}(a_1, a_2) \\ d_{eucl}(a_1, a_3) \end{cases} & \text{otherwise} \end{cases} \tag{23}$$

where c is the cost parameter that used when there is a gap in MTS alignment.

3.3. Geometry-based measures

This kind of measures uses the shape of sequences as a geometric feature of the MTS.

3.3.1. Hausdorff

The Hausdorff (Besse, et al., 2016; Hausdorff, 1927) distance shows the spatial similarity between two MTSs that measures how MTSs are far from each other. If every sample of either MTS is close to some other MTS samples, then the Hausdorff distance is low. The conventional Hausdorff considers not only the sampling point, but also every point of the sequences, thus it is a complicated measure. The simple version of it, was proposed based on point to segment distance that defined by:

$$D_{Hausdorff}\left(A_1^p, B_1^q\right) = max \left\{ \begin{array}{l} \underset{a_i \in A_1^p}{max} \left\{ \underset{bs_j \in Bs_1^q}{min} \left\{ d_{p2s}\left(a_{1,i}, bs_{2,j}\right) \right\} \right\} \\ \underset{b_i \in B_1^q}{max} \left\{ \underset{as_j \in As_1^p}{min} \left\{ d_{p2s}\left(b_{2,i}, as_{2,j}\right) \right\} \right\} \end{array} \right. \qquad (24)$$

3.3.2. *Fréchet*

The Fréchet (Fréchet, 1906) distance considers the data samples with their orders along the continuous sequences. The shortest distance that needs to connect is the Fréchet distance between two MTSs. Indeed, the Fréchet distance between segments is equal to the Hausdorff distance between MTS segments and it is the resource intensive measure. Eiter et al. (Eiter & Mannila, 1994) presented the discrete Fréchet (disFréchet) distance to approximate the exact Fréchet distance efficiently based on the recursive model. This method reduces the complexity of discrete Fréchet distance to $O(p.q)$ as follows:

$$D_{disFréchet}\left(A_1^p, B_1^q\right) = max \left\{ \begin{array}{l} min \left\{ \begin{array}{c} D_{disFréchet}\left(A_1^{p-1}, B_1^{q-1}\right) \\ D_{disFréchet}\left(A_1^{p-1}, B_1^{q}\right) \\ D_{disFréchet}\left(A_1^{p}, B_1^{q-1}\right) \end{array} \right. \\ \qquad\quad d_{eucl}\left(a_p, b_q\right) \end{array} \right. \qquad (25)$$

3.3.3. *SSPD*

The Symmetric Segment Path Distance (SSPD) (Besse, et al., 2016) is dependent on the point-to-segment distance like the Hausdorff. The SSPD distance computes the minimum point-to-segment distance for every point of the first MTS in all segments of the other one. Afterward, the average of the computed distance of every MTS sample is reported as SSPD distance.

$$D_{SSPD}\left(A_1^p, B_1^q\right) = \frac{1}{2p}\sum_{i=1}^{p} \min_{bs_j \in Bs_1^p} d_{p2s}\left(a_i, bs_j\right) + \frac{1}{2q}\sum_{i=1}^{q} \min_{as_j \in As_1^p} d_{p2s}\left(b_i, as_j\right) \tag{26}$$

where p and q are the length of given MTSs.

3.4. Differential-based measures

The first order difference of MTS is the basis of similarity measures in this category. The differential-based methods use most of times the DTW as a base measure.

3.4.1. CIDTW

Batista et al. (Batista, Keogh, Tataw, & De Souza, 2014) presented the CIDTW similarity measure as a weighting method to compensate the complexity difference of two comparing MTS with the summation squares of the first-order difference. This weight is a multiplicative complexity-based value that weighs the DTW distance.

$$D_{CIDTW}\left(A_1^p, B_1^q\right) = D_{DTW}\left(A_1^p, B_1^q\right) \times \frac{max\begin{cases} complexity\left(A_1^p\right) \\ complexity\left(B_1^q\right) \end{cases}}{min\begin{cases} complexity\left(A_1^p\right) \\ complexity\left(B_1^q\right) \end{cases}} \tag{27}$$

$$complexity\left(A_1^p\right) = \sqrt{\sum_{i=1}^{p-1}\left(a_i - a_{i+1}\right)^2} \tag{28}$$

The CIDTW was an attempt to correct for the bias towards finding everything relatively similar to a simple shape

3.4.2. DDTW

The DDTW similarity measure proposed as a derivative-based distance that is a weighted combination of the DTW distance of raw MTS with the first-order MTS differences (Górecki & Łuczak, 2013; E. J. Keogh & Pazzani, 2001).

$$D_{DDTW}\left(A_1^p, B_1^q\right) = cos\alpha \times D_{DTW}\left(A_1^p, B_1^q\right) + sin\alpha \times D_{DTW}\left(diff\left(A_1^p\right), diff\left(B_1^q\right)\right) \qquad (29)$$

where α is the weighting parameter that could highly affect the method efficiency.

The DDTW was tried to improve the DTW by considering the differential feature of time series shape, the trend of time series samples, in addition to the only distance features that used by DTW measure.

A summary of the mentioned similarity measures is shown in Table 1. The first column shows the category of similarity measures; the second column is the method name with references. Methods parameters are presented in the third column. In the fourth and fifth columns, distance type and time complexity are given, respectively.

Table 1 A summary of MTS similarity measures

Category	Method [Ref]	Free parameters	Type	Time Complexity
Elastic measures	DTW (Berndt & Clifford, 1994)	-	symmetric	$O(p.q)$
	cDTW (Sakoe & Chiba, 1978)	δ	symmetric	$O(max(p,q).\delta)$
	WDTW (Jeong, et al., 2011)	g	symmetric	$O(p.q)$
	LCSS (Vlachos, et al., 2002)	ϵ	distance	$O(p.q)$
	EDR (Chen & Ng, 2004)	ϵ	symmetric	$O(p.q)$

	ERP (Chen, et al., 2005)	gap	metric	$O(p.q)$
	SWALE (Morse & Patel, 2007)	ϵ, r_m, g_c	symmetric	$O(p.q)$
	TWED (Marteau, 2009)	υ, λ	metric	$O(p.q)$
	MSM (Stefan, et al., 2013)	c	symmetric	$O(p.q)$
Geometry-based measures	Hausdorff (Hausdorff, 1927)	-	metric	$O(p.q)$
	disFréchet (Eiter & Mannila, 1994)	-	symmetric	$O(p.q)$
	SSPD (Besse, et al., 2016)	-	symmetric	$O(p.q)$
Differential-based measures	CIDTW (Batista, et al., 2014)	-	symmetric	$O(p.q)$
	DDTW (Górecki & Łuczak, 2013)	α	symmetric	$O(p.q)$

4. Similarity measures evaluation framework

In this section, it will be shown that how the similarity measures have been evaluated on MTS datasets.

4.1. Computation time

The time required to compute the distance matrix for all MTS datasets is calculated as a criterion to compare the computation cost between different similarity measures.

4.2. Classification design

In the context of time-series, the efficiency of similarity measures was addressed by using a simple 1-Nearest Neighbor (1-NN) classifier on labelled MTS datasets (Serra & Arcos, 2014; X. Wang, et al., 2013). Each MTS data has a correct class label, and the classifier tries to predict the label as that of its nearest neighbor in the training set (Salzberg, 1997). There are several advantages with choosing 1-NN method. First, the 1-NN classifier performance is affected directly with the underlying distance, hence, the accuracy of the classifier is clearly reflects the effectiveness of the similarity measure. Second, the 1-NN classifier is a parameter-free method which makes it easy to reproduce the results straightforwardly. Finally, to the best of our knowledge, the 1-NN classifier is competitive and hard to beat (Ding, et al., 2008; Serra & Arcos, 2014; X. Wang, et al., 2013). Also, the precision criterion (Powers, 2011) is used to evaluate the classification result on testing set as follows:

$$P = \frac{TP}{TP+FP} \times 100 \tag{30}$$

where P is the precision, TP is the number of MTSs that are correctly classified and FP is the MTSs that are incorrectly classified.

A k-fold technique with balanced labels, as a standard tool to generalize the results, was used in this paper for properly assessing a classifier's accuracy (Abu-Mostafa, Magdon-Ismail, & Lin, 2012). The stratified cross-validation estimates the model efficiency regarding the class distribution and prevent bias arising from a particular data partitioning (Abu-Mostafa, et al., 2012). If the similarity measure required parameter tuning, the training set was divided into two equal size and one of the subset for parameter tuning by leave-one-out classification test was used. Furthermore, repeating the cross-validation decreases the variance of the results. The procedure of 1-NN classification through k-fold is shown in Algorithm 1 (Ding, et al., 2008; X. Wang, et al., 2013).

Algorithm 1 MTS classification with 1-NN classifier

```
Input:      Labeled MTS dataset $X$
            The number of folds $kf$
            The number of replications $rep$
Output:     Average 1-NN classification accuracy

Let $Res$ be the $kf \times rep$ matrix
for $i \leftarrow 1$ to $rep$
    Randomly partition $X$ into $kf$ stratified subsets $X_1 \ldots X_{kf}$
    for $X_j \leftarrow X_1$ to $X_{kf}$ do
        if similarity measure requires parameter tuning then
            Randomly partition $X_j$ into two equal sized stratified subsets $X_{j1}$ and $X_{j2}$
            Use $X_{j1}$ for parameter tuning using leave-one-out cross-validation
            Set the parameter to value that yields maximum classification accuracy
        Use $X_j$ as a training set, $X - X_j$ as testing set
        $Res[i, j] \leftarrow$ the 1-NN classification accuracy
return Average of $Res$ matrix
```

Where X is an MTS dataset, kf is the number of folds for stratified cross-validation, and rep is the number of repetition to get more precise results.

4.3. Parameter tuning

Several investigated similarity measures have one or more controlling parameter that choosing these parameters directly affects the measures productivity (Serra & Arcos, 2014). In this experiment, the grid search within a suitable range of parameters was used as a possible values of the parameters that can be chosen according to the given specification in the introducing papers of each measure, as described in Table 2.

Table 2 Parameter grid for the considered similarity measures (recall that n_1 and n_2 corresponds to the length of the input MTS).

Measure	Parameter	Min	Max	No. of Steps
cDTW (Sakoe & Chiba, 1978)	Windows size - δ	$\lvert p - q \rvert$	$0.3 \times max(p, q)$	30
WDTW (Jeong, et al., 2011)	Curvature - g	0	1	5

LCSS (Vlachos, et al., 2002)	Distance threshold - ϵ	$0.02.std(X)$	$std(X)$	5
EDR (Chen & Ng, 2004)	Distance threshold - ϵ	$0.02.std(X)$	$std(X)$	5
ERP (Chen, et al., 2005)	Penalty point - gap	0	$\mp 3.std(X)$	3
SWALE (Morse & Patel, 2007)	Match reward - r_m	$50std(X)$		-
SWALE (Morse & Patel, 2007)	Penalty cost - g_c	0	r_m	5
SWALE (Morse & Patel, 2007)	Distance threshold – ϵ	$0.02.std(X)$	$std(X)$	5
TWED (Marteau, 2009)	Stiffness - ν	10^{-4}	10^{0}	5
TWED (Marteau, 2009)	Penalty - λ	0	$std(X)$	5
MSM (Stefan, et al., 2013)	Cost - c	$\{10^{-2},10^{-1},10^{0},10^{1},10^{2},10^{3},10^{4},10^{5}\}$		8
DDTW (Górecki & Łuczak, 2013)	Ratio - α	1	$\pi/2$	5

For cDTW similarity measure, the windows size δ is an optional parameter, the best warping window size is searched in the linearly-spaced integer values from difference length of compared MTSs to the 30% of the maximum length of given MTSs. For WDTW the 5 linearly-spaced real values from 0 to 1 was used. For distance threshold ϵ parameter, that is common in LCSS, EDR, and SWALE, the linearly-spaced real values from 2% of MTS standard deviation up to the full value of MTS standard deviation was searched. For SWALE the reward parameter was fixed to 50 times of MTS standard deviation and the optimal value for penalty cost was searched from 0 to the reward value. For TWED all 25 possible combination of $\nu \in [10^{-4},10^{0}]$ and $\lambda \in [0,std(X)]$ was used. For the MSM measure all values $c = \{10^{-2},10^{-1},10^{0},10^{1},10^{2},10^{3},10^{4},10^{5}\}$ was searched to find the optimal value of the cost parameter. The α parameter of the DDTW measure was tuned from 1 to $\pi/2$. For each similarity measure, the parameter value yielding to the highest leave-one-out accuracy for the training set was kept for out-of-sample testing.

4.4. Statistical significance

The Wilcoxon signed rank test (Hollander, Wolfe, & Chicken, 2013; Wilcoxon, 1992) was applied to evaluate the statistical significance of the difference between the two measures accuracy. The Wilcoxon signed sank test is a non-parametric analysis that statistically compares the average of two repeated measurements and assesses for significant differences. It is equivalent to dependent Student's t-test when it does not assume normality in the data. For each MTS dataset the $rep \times kf$ accuracy values obtained for each classifier was used as an input for the Wilcoxon singed rank test to compare statistically the best performer with the second best measure. Besides, the non-parametric rank-based test is an accepted statistic for comparing the performance of n_{cl} classifier over n_X datasets (Demšar, 2006; García, Fernández, Luengo, & Herrera, 2010). A null hypothesis assumes that the average performance rank of n_{cl} similarity measures on n_X MTS datasets are the same (not significantly different). There is an alternative hypothesis against the null hypothesis which assumes at least one classifier's mean rank is different. Hence the matrix M (n_{cl} by n_{cl}) includes the accuracy of similarity measures. At the first stage the performance rank of every n_{cl} classifier was evaluated for each dataset separately and made an R matrix, where r_{ij} is the rank of the j^{th} classifier on the i^{th} dataset. The rank of classifiers with the equal accuracy wereare averaged. The average rank of each classifier is denoted as $R_j = \frac{1}{n_X}\sum_i r_{ij}$. Under the null hypothesis, the ranks of all classifiers is equal, the Friedman statistics (Friedman, 1940), F_F, can be approximated by F-distribution with $(n_{cl} - 1)$ and $(n_{cl} - 1)(n_X - 1)$ degree of freedom as it follows:

$$F_F = \frac{(n_X - 1)\chi_F^2}{n_X(n_{cl} - 1) - \chi_F^2}$$
$$\chi_F^2 = \frac{12.n_X}{n_{cl}(n_{cl} + 1)}\left[\sum_j R_j^2 - \frac{n_{cl}(n_{cl} + 1)^2}{4}\right] \qquad (31)$$

where F_F is the Friedman statistics value. If the null hypothesis is rejected based on the test results, the further family-wise comparisons will be needed. The Holm (Holm, 1979) step-down post-hoc method was used to compensate multiple family-wise comparisons. Demšar (Demšar, 2006) recommends grouping classifiers into cliques, within which there is no significant difference in rank. This allows the average ranks and groups of not significantly different classifiers to be plotted on an order line in a graph referred to as a critical difference diagram.

5. Experimental Evaluation

In this section, the effectiveness of 14 similarity measures include: DTW, cDTW, WDTW, LCSS, EDR, ERP, SWALE, TWED, MSM, Hausdorff, disFréchet, SSPD, DDTW, and the CIDTW were evaluated over 32 publicly-available datasets. The results of the 1-NN classifier with each of mentioned similarity measures were compared**.** The entire simulation was conducted on a CORE-I7 computer with 16GB of RAM running for over a month.

5.1. Datasets

Experiments were performed using 32 publicly-available labeled MTS datasets with varying properties that are presented briefly in Table 3. They include synthetic, as well as real-world datasets. There are some criteria to characterize the datasets such as the number of time series, the average length of time series, and the average shape complexity (ASC) (Zhang, Huang, & Tan, 2006) as mentioned in Table 3. The shape complexity can be calculated as follows:

$$\xi_{A_1^p} = \frac{d_{eucl}(a_1, a_p)}{\sum_{i=1}^{p-1} d_{eucl}(a_i, a_{i+1})} \qquad (32)$$

where $\xi_{A_1^p}$ is the shape complexity of time series A_1^p.

Table 3 Summary of datasets

	Size	#Class	#Variables	Min length	Max length	ASC
'ASL-10' (Vlachos, et al., 2002)	699	10	2	33	296	0.03
'ASL-35' (Kadous, 1995)	700	35	2	30	220	0.02
'VMT' (Hu, Li, Tian, Maybank, & Zhang, 2013)	1500	15	2	16	612	0.87
'SM' (Hu, et al., 2013)	2500	50	2	160	1851	0.25
'CROSS'(Morris & Trivedi, 2009)	1900	19	2	5	23	0.81
'I5' (Morris & Trivedi, 2009)	806	8	2	4	27	1.00
'I5SIM1' (Morris & Trivedi, 2009)	800	8	2	14	22	0.81
'I5SIM2' (Morris & Trivedi, 2009)	1600	8	2	14	141	0.60
'I5SIM3' (Morris & Trivedi, 2009)	1600	16	2	14	141	0.60
'LABOMNI' (Morris & Trivedi, 2009)	209	15	2	30	624	0.49
'FT' (Beyan & Fisher, 2013)	3102	2	2	2	82	0.61
'HC-digit'(Ramos-Garijo, et al., 2007)	198	9	2	8	110	0.47
'HC' (Ramos-Garijo, et al., 2007)	1363	35	2	4	155	0.47
'CAL1' (Calderara, Prati, & Cucchiara, 2011)	400	2	2	60	60	0.88
'CAL2' (Calderara, et al., 2011)	670	3	2	60	60	0.88
'CAL3' (Calderara, et al., 2011)	900	4	2	60	60	0.88
'CAL4' (Calderara, et al., 2011)	1210	5	2	60	60	0.79
'CAL5' (Calderara, et al., 2011)	1130	8	2	60	60	0.75
'CAL6' (Calderara, et al., 2011)	380	3	2	60	60	0.76
'CAL7' (Calderara, et al., 2011)	220	3	2	60	60	0.95
'CAL8' (Calderara, et al., 2011)	120	18	2	99	99	0.85
'CAL9' (Calderara, et al., 2011)	300	4	2	60	60	0.66
'SIGNATURE' (Yeung, et al., 2004)	1600	40	2	91	793	0.20
'ArabianDigit' (Hammami & Bedda, 2010)	8800	10	13	4	93	0.07
'ASL2Full' (Kadous, 1995)	2565	96	22	45	136	0.28
'CharTraj' (Williams, Toussaint, & Storkey, 2007)	2858	20	3	109	205	0.00
'Japanese Vowels' (Kudo, Toyama, & Shimbo, 1999)	640	9	12	7	29	0.50
'robotLP1' (Lopes & Camarinha-Matos, 1998)	88	4	6	15	15	0.41
'robotLP2' (Lopes & Camarinha-Matos, 1998)	47	5	6	15	15	0.17
'robotLP3' (Lopes & Camarinha-Matos, 1998)	47	4	6	15	15	0.17
'robotLP4' (Lopes & Camarinha-Matos, 1998)	117	3	6	15	15	0.26
'robotLP5' (Lopes & Camarinha-Matos, 1998)	164	5	6	15	15	0.24

5.2. Classification performance

In our experiment the standard 3-fold cross validation (kf = 3) was followed for the classification task. The smaller training set (larger the number of folds) is used for cases that the several similarity measures achieved perfect classification performance. Also, the validation was repeated 50 times (rep = 50) and the accuracy was averaged. The classification performance of the discussed similarity measures along with the average rank of them for all datasets are given in Table 4. Every column includes the accuracy of the considered similarity measure for each MTS dataset. The best performance over each MTS dataset was bolded and the measures which statistically significantly outperformed the other measures were indicated by the star (*). The last row shows the average rank of each similarity measures that is the average position after sorting the accuracy for a given data set in descending order.

It is observed that all similarity measures achieved perfect or near-perfect accuracies for five datasets (I5SIM1, I5SIM2, CAL1, CAL2, and CAL3) with 3-fold cross validation, hence the experiment is run again with larger number of folds for those datasets. However, no single measure achieved the best performance for all the datasets. The DDTW method was found to be the best-performing in 13 datasets, Hausdorff in 7 datasets, CIDTW and WDTW in 6 datasets, SSPD and disFréchet in 5 datasets, MSM and DTW in 2 datasets, and cDTW works the best in 1 dataset. For three datasets more than one similarity measure shows the perfect best performance. If only count the datasets where one measure statistically significantly outperforms the rest , the numbers reduce to 10 for DDTW, 4 for CIDTW and WDTW, 2 for SSPD, and 1 for DTW, MSM, Hausdorff, and disFréchet. Hence, there are some datasets that choosing a specific similarity measure can make a difference in performance.

However, some dataset attributes can be highlighted based the measures accuracy. For instance, the fact that the specific similarity measure outperforms the others for some datasets indicates that similar kind of datasets may be well characterized by the aforementioned similarity measure (e.g., MSM with I5SIM3, DDTW with HC, CIDTW with robotLP1m and SSPD with robotLP3).

Table 4 Accuracy (%) for all considered measures and MTS datasets. Every column includes accuracy. The last row contains the average rank of each measure across all datasets. The best performances for each dataset are bolded.

Datasets	**Similarity measure**													
	'DTW'	**'cDTW'**	**'WDTW'**	**'LCSS'**	**'EDR'**	**'ERP'**	**'SWALE'**	**'TWED'**	**'MSM'**	**'Hausdorff'**	**'disFrechet'**	**'SSPD'**	**'CIDTW'**	**'DDTW'**
'ASL-10'	78.11	77.70	**80.22***	39.04	63.82	76.89	41.99	78.91	78.51	70.20	72.09	73.28	79.99	79.24
'ASL-35'	25.53	25.82	25.53	14.45	19.99	23.12	14.53	21.87	24.46	18.61	20.53	19.85	26.33	**27.23***
'VMT'	96.52	96.54	96.52	95.28	94.95	88.92	95.38	94.15	96.58	93.05	93.08	**96.99***	96.49	96.53
'SM'	97.16	97.24	97.16	37.94	96.99	94.89	82.08	96.75	96.63	77.58	98.34	73.74	97.31	**98.62***
'CROSS'	99.49	99.52	**99.67***	84.68	94.41	99.66	88.55	99.65	99.67	99.62	99.52	99.18	99.51	99.19
'I5'	97.33	97.18	97.33	99.56	97.82	89.20	**99.62**	98.25	98.26	92.71	91.80	98.88	96.82	96.36
'I5SIM1'	**83.21**	**83.21**	**83.21**	82.96	78.83	76.31	82.96	83.20	**83.21**	**83.21**	**83.21**	**83.21**	**83.21**	**83.21**
'I5SIM2'	99.69	99.69	99.69	99.94	75.50	52.71	99.94	89.76	97.14	**100**	**100**	**100**	98.62	88.43
'I5SIM3'	69.42	69.42	99.85	21.25	94.41	99.53	50.00	99.99	**100***	84.74	90.30	55.98	89.11	88.37
'LABOMNI'	94.49	94.10	94.49	86.83	85.18	76.86	88.22	86.82	89.96	68.95	**96.08***	74.29	94.03	94.42
'FT'	98.17	98.14	98.26	97.98	98.07	98.05	97.30	98.11	98.09	97.69	98.24	97.57	**98.40***	98.18
'HC-digit'	75.05	75.60	77.09	30.02	72.16	62.15	52.84	72.21	65.16	53.63	73.09	48.76	71.88	**83.08***
'HC'	54.22	54.30	55.45	7.70	50.98	42.84	27.06	51.35	45.64	41.92	55.18	35.69	53.10	**64.21***
'CAL1'	**97.96**	**97.96**	**97.96**	97.59	97.43	**97.96**	97.59	**97.96**	**97.96**	**97.96**	**97.96**	**97.96**	**97.96**	**97.96**
'CAL2'	98.17	98.17	98.17	98.17	98.17	98.17	98.17	98.17	98.02	**98.17**	98.17	98.17	98.17	98.17
'CAL3'	98.07	98.07	98.07	98.06	98.06	98.07	98.06	97.96	98.04	**98.07**	**98.07**	98.07	98.07	98.07
'CAL4'	99.51	99.53	99.49	99.26	99.21	99.51	99.26	99.51	99.48	99.56	99.57	99.43	99.49	**99.80***
'CAL5'	96.97	97.25	97.26	85.23	84.53	96.97	85.23	97.16	97.12	97.78	97.91	96.54	97.20	**97.95**
'CAL6'	99.51	99.45	99.33	99.39	99.42	99.51	99.39	99.51	99.46	99.27	99.19	99.22	99.33	**99.73***
'CAL7'	97.92	98.11	98.11	90.58	90.05	97.92	90.58	97.90	97.93	**98.78**	98.77	98.07	98.11	98.30
'CAL8'	68.06	68.06	69.29	65.76	65.84	68.08	65.76	67.61	69.40	68.34	68.15	68.27	69.29	**70.70***
'CAL9'	95.62	96.71	96.84	95.22	95.06	95.62	95.22	95.78	95.76	97.85	98.09	96.26	96.84	**100***
'SIGNATURE'	90.93	90.96	90.97	49.77	92.22	85.79	58.11	91.79	91.11	85.61	94.36	82.10	92.06	**99.70***

'ArabianDigit'	96.36	96.38	**96.87***	92.30	96.56	96.14	94.87	94.97	95.29	46.88	82.56	62.78	95.50	94.73
'ASL2Full'	64.12	64.21	64.14	56.57	63.51	57.59	62.76	56.73	57.67	19.91	64.70	15.80	62.03	**73.28***
'CharTraj'	**99.75***	97.63	97.73	77.93	98.76	99.07	90.85	99.35	99.74	88.00	96.05	89.11	97.63	97.78
'Japanese Vowels'	93.66	93.86	**94.94***	93.04	94.03	92.71	9.53	93.75	93.42	62.66	93.52	62.93	92.70	93.33
'robotLP1'	76.56	79.76	79.80	53.01	51.46	80.61	53.01	76.63	76.56	67.83	84.39	61.37	**86.24***	77.68
'robotLP2'	53.23	54.64	55.20	51.80	51.50	53.21	51.80	52.42	54.64	**59.59***	53.96	56.10	53.22	55.22
'robotLP3'	55.90	56.67	59.32	58.13	54.85	56.24	58.13	56.14	56.28	65.77	59.79	**67.19***	63.29	60.01
'robotLP4'	82.70	84.04	84.77	56.53	50.35	87.43	56.53	82.64	82.68	86.32	86.76	75.89	**89.19***	78.11
'robotLP5'	60.95	64.35	64.20	42.32	44.18	64.99	42.32	60.88	60.94	52.46	62.11	53.37	**67.09***	58.64
Average rank	6.64	5.72	4.61	11.34	10.09	8.27	10.84	7.75	6.92	8.02	5.33	9.22	5.75	**4.50**

* A statistically significant difference with respect to the other measures for a given dataset ($p < 0: 05$, Section 4.3.)

In a more global scale, average ranks by themselves provide a useful comparison of the algorithms. On average, the DDTW technique, that showed the best performance for 10 datasets, ranked the first with rank 4.5; WDTW and disFréchet ranked the second and the third, with ranks 4.61 and 5.33, respectively. The Friedman test proves whether the measured average ranks are significantly different from the mean rank $R_j = 7.5$ expected under the null hypothesis: $Friedman(ImanDavenport)F_F = 12.65$ with 14 algorithms and 32 datasets, F_F is distributed according to the F-distribution with $14 - 1 = 13$ and $(14 - 1).(32 - 1) = 403$ degrees of freedom. The p-value computed by using the $F(13{,}403)$ distribution is 2.37×10^{-23}, so the null hypothesis was rejected at a high confidence level.

Demsar (Demšar, 2006) recommends grouping classifiers into cliques, within which there is no significant difference in rank. This allows the average ranks and groups of not significantly different classifiers to be plotted on an order line in a graph referred to as a critical difference diagram. In this way, Fig. 3 shows the critical statistical difference diagram for 14 similarity measures over the 32 datasets. As shown in Fig. 3 there is no one similarity measure that significantly outperforms the others. There are four cliques within which no significant difference is observed. The top clique contains all but SSPD, LCSS, EDR, and SWALE similarity measures. It means there is no significantly difference between similarity measures (other than SSPD, LCSS, EDR, and SWALE) for the 1-NN classification task based on our MTS datasets. These results do not lend any support to each of similarity measure over other MTS similarity measures.

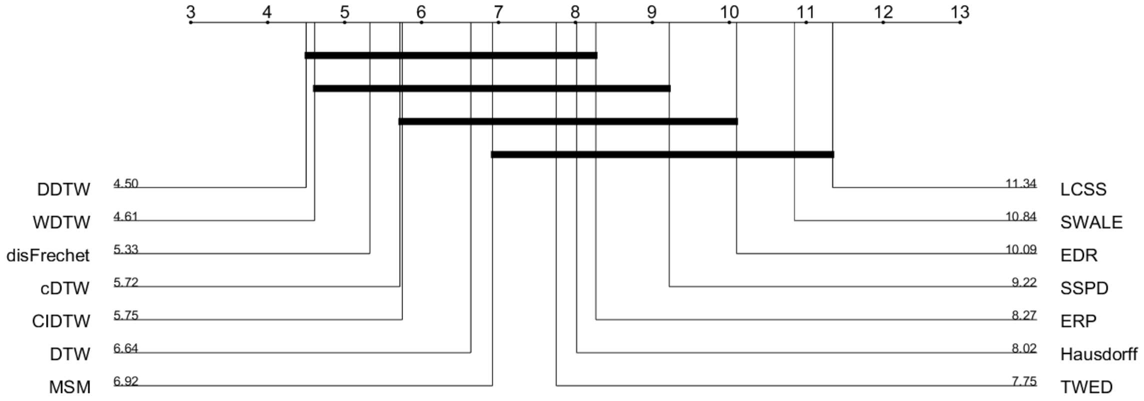


Fig. 3 The average ranks for 14 similarity measures over 32 datasets for $\alpha = 5\%$.

Fig. 4 presents the behavior of DDTW, as the best measure based on average rank, versus each competitor based on the number of datasets where DDTW produces respectively better performance, equal performance, and worse performance compared to each of them. The goal of this experiment has been to compare the competitive performance of DDTW based on 1-NN classification compared to other similarity measures. As can be seen from Fig. 4, the WDTW and disFréchet measures have the lowest number of datasets that the DDTW works better than. In the opposite side, the LCSS technique has the highest number of datasets where theirs performance is worse than DDTW similarity measure.

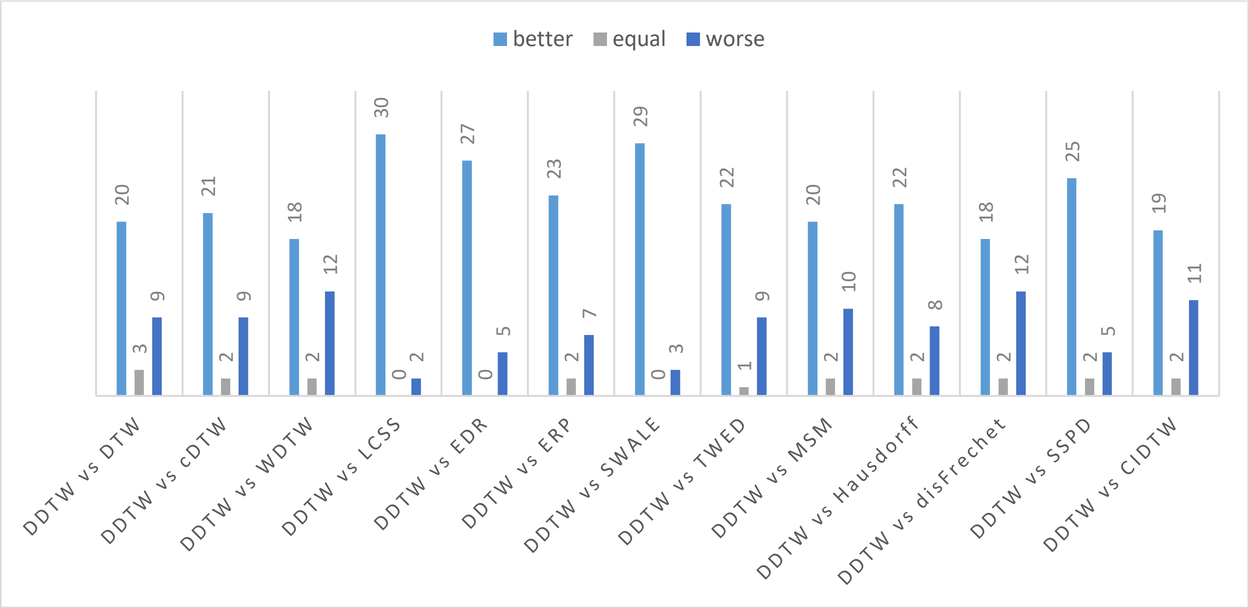
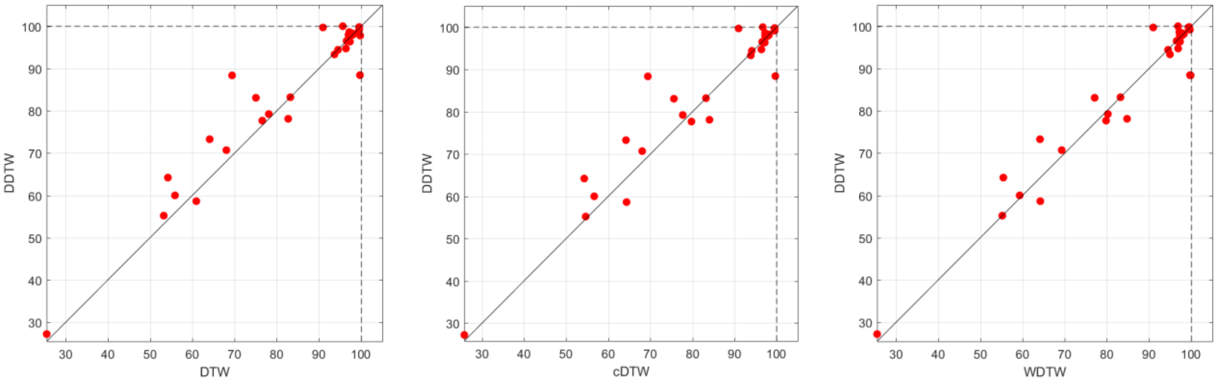


Fig. 4 The number of datasets which DDTW produced better, equal, or worse performance compared to other similarity measures.

To provide a more intuitive illustration of the performance of the similarity measures compared in Table 4, scatter plots were utilized to conduct pair-wise comparisons. In a scatter plot, the accuracy of the two similarity measures under comparison were used as the x and y coordinates of a dot, where each dot represents a particular dataset. Where a scatter plot has the label "A versus B", a plot above the line indicates that A is more accurate than B. The further a dot is from the line, the greater the margin of accuracy improvement. The more dots on one side of the line indicates that the better one similarity measure is compared to the other.

To provide a more intuitive illustration of the performance of the similarity measures compared to the DDTW as the best performer on our MTS datasets based on the average rank, the pairwise comparison conducted through the scatter plot was used. In a scatter plot, the accuracy of the DDTW was used as the y coordinate of a dot and the accuracy of the similarity measure under comparison was used as$x$ coordinates of a dot, where each dot represents a particular dataset. Each scatter plot has the label “DDTW versus A”, a plot above the line indicates that DDTW is a better performer than A (X. Wang, et al., 2013). The further a dot is from the line, the greater the margin of the accuracy improvement. The more dots on one side of the line indicates that the better one similarity measure is compared to the other.

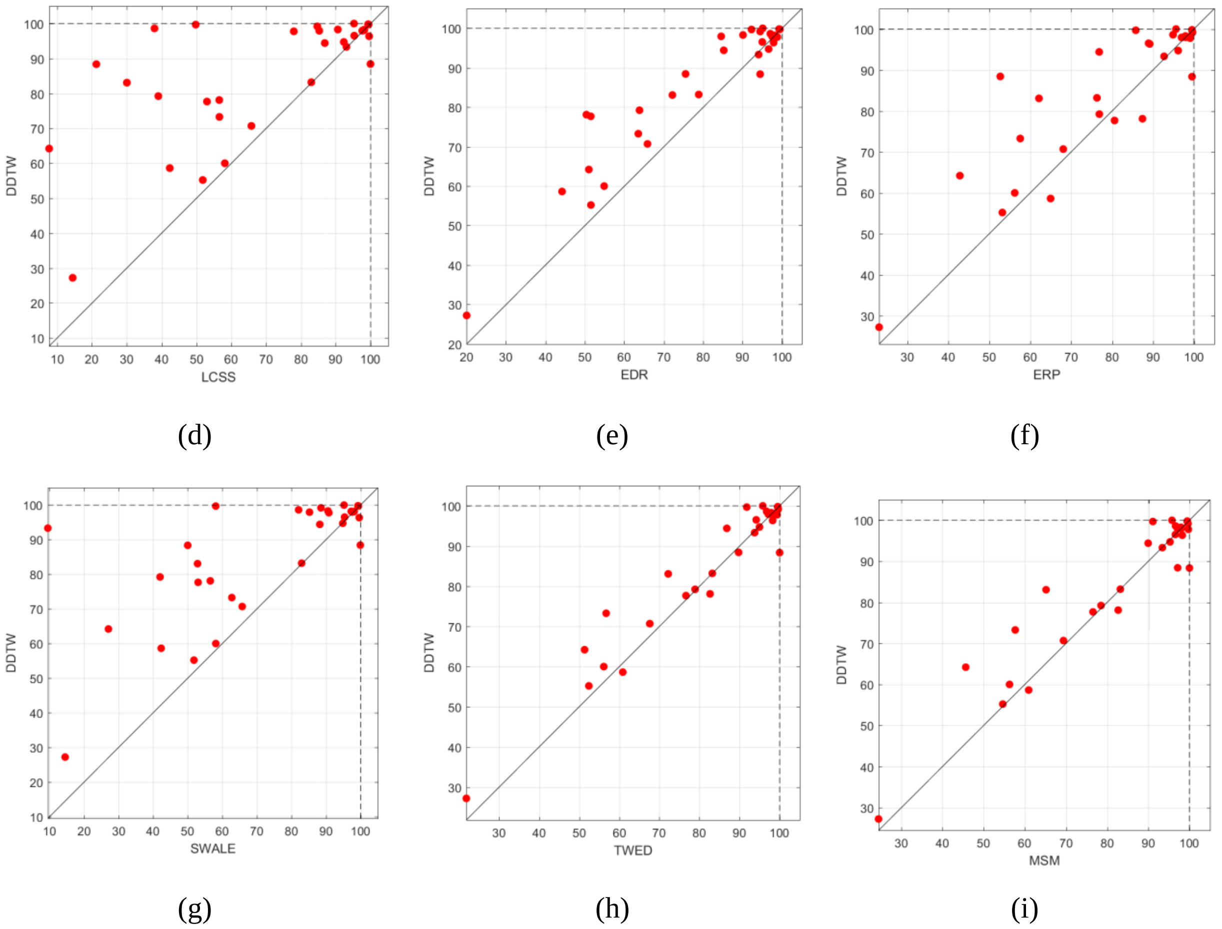
The performance of DDTW against its elastic competitors is shown in Fig. 5. From Fig. 5 (a,b), it can be seen that the effectiveness of DDTW is better than that of DTW and cDTW. It can be observed in Fig. 5 (c) that the accuracy of DDTW is slightly better than WDTW measure. As can be seen from Fig. 5 (d, e, and g) the DDTW measure is clearly superior over LCSS, EDR, and SWALE measures. An obvious conclusion from Fig. 5 (f, h, and i) is that DDTW measure outperforms these three similarity measures (ERP, TWED, and MSM).



(a)

(b)

(c)

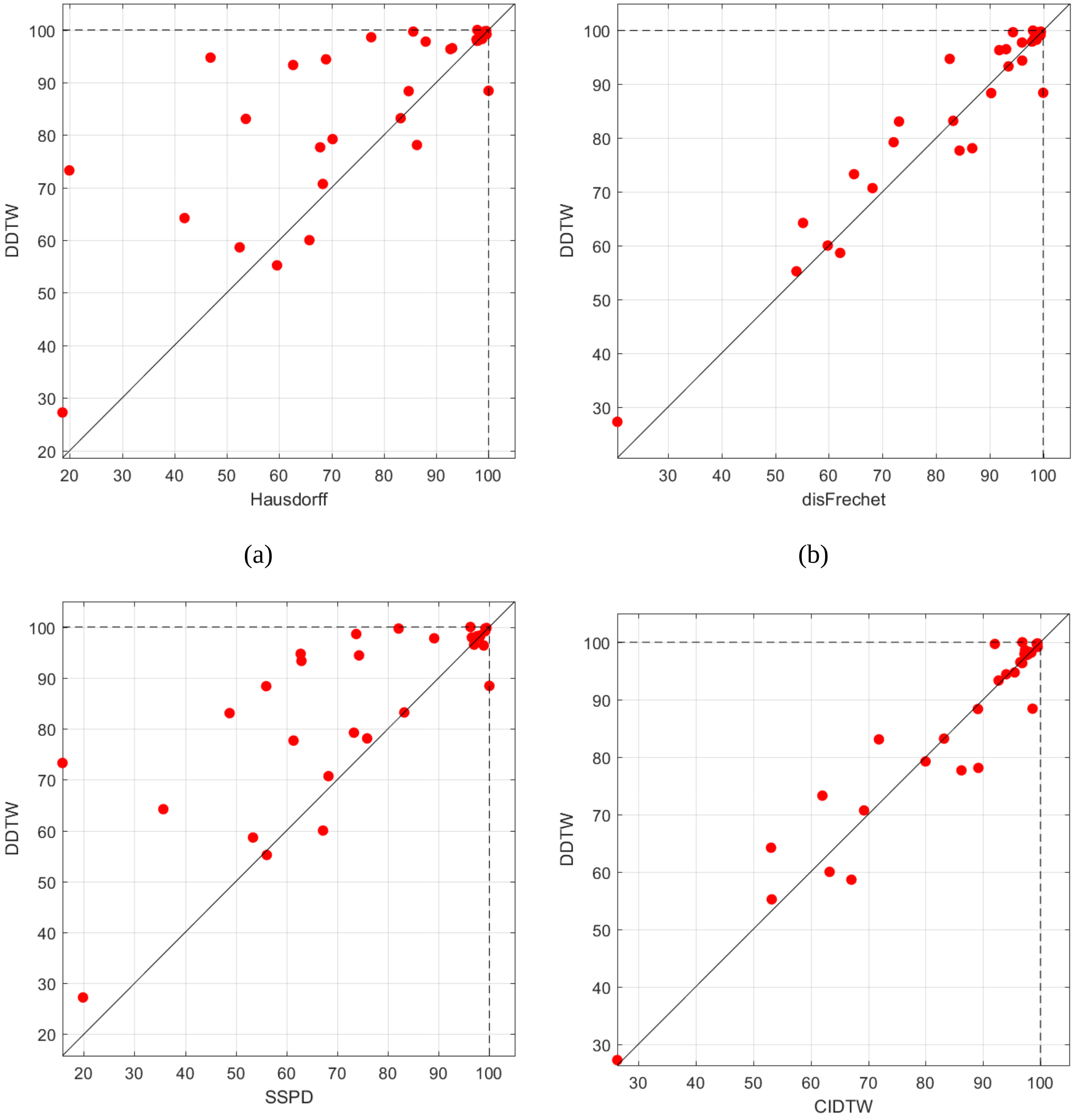


(d) (e) (f)

(g) (h) (i)

Fig. 5 Pairwise comparison of DDTW against elastic similarity measures. (a) DDTW vs DTW, (b) DDTW vs cDTW, (c) DDTW vs WDTW, (d) DDTW vs LCSS, (e) DDTW vs EDR, (f) DDTW vs ERP, (g) DDTW vs SWALE, and (h) DDTW vs TWED, (i) DDTW vs MSM.

Fig. 6 depicts the performance of DDTW against its geometry-based and differential-based competitors. As shown in Fig. 6 (a and c) DDTW measure is clearly superior to Hausdorff and SSPD similarity measures on tested datasets. It can be observed in Fig. 6 (b and d) that the effectiveness of DDTW is slightly better than that of disFréchet and CIDTW measures.



(a)

(b)

(c)

(d)

Fig. 6 Pairwise comparison of DDTW against geometry-based and differential-based similarity measures. (a) DDTW vs Hausdorff, (b) DDTW vs disFréchet, (c) DDTW vs SSPD, and (d) DDTW vs CIDTW.

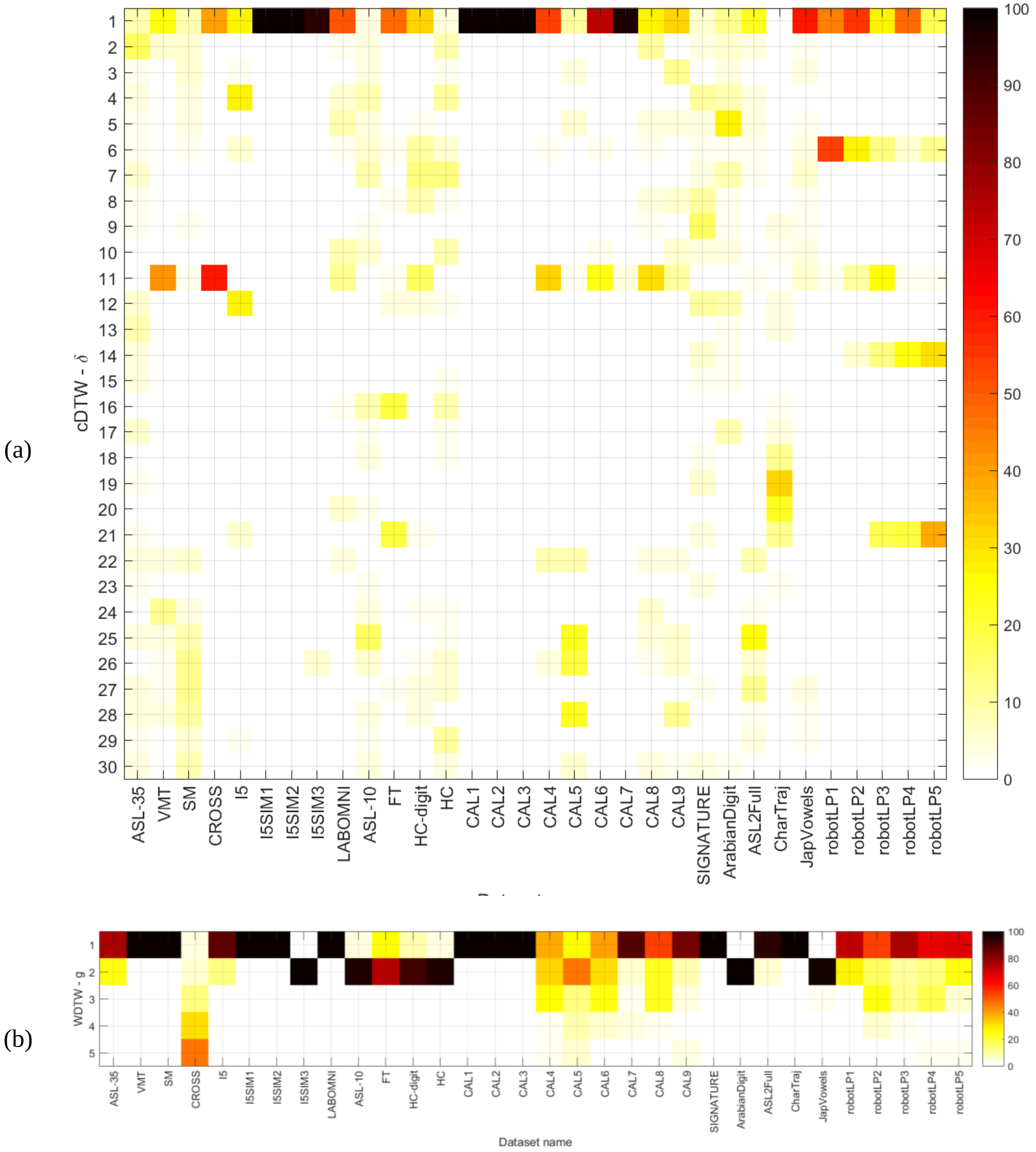
In summary, we found through experiments that there is no clear evidence that one similarity measure exists that is clearly superior to others in the literature in terms of accuracy. There are some similarity measures that are more effective on certain datasets, but inferior on some other datasets. However, based-on the experiments and used MTS datasets, DDTW measure totally show the best performance among all similarity measures in terms of average rank and number of best performing datasets.

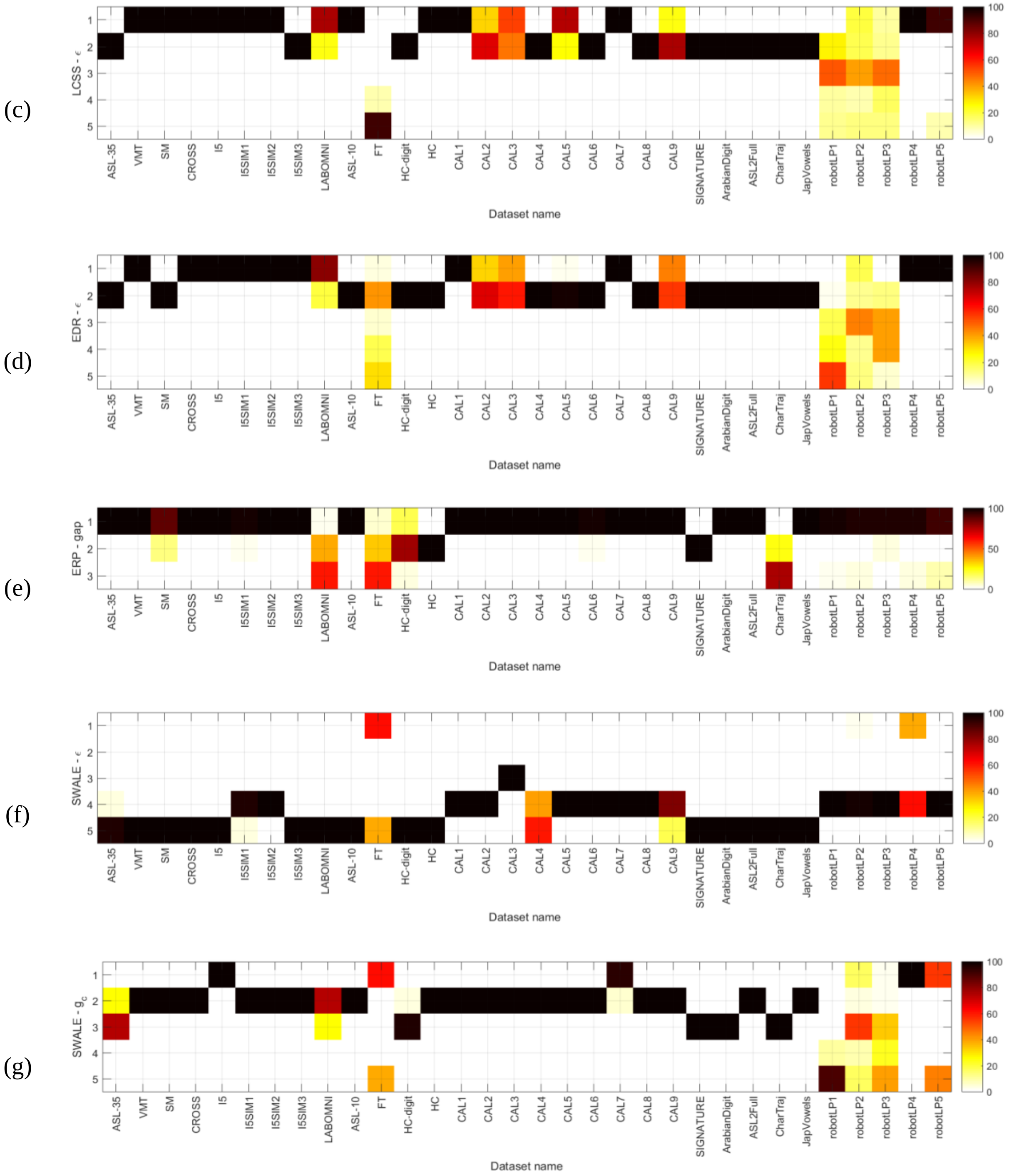
5.3. Parameter assessment

To select the proper parameters for every method, 50% of dataset train data was used for each of 50 replications. The chosen parameters for each similarity measure is reported in Fig. 7. Each color in Fig. 7 shows the percentage of times that a given parameter value was chosen for each dataset. The color black determines the particular parameter was chosen for each dataset in all iterations (Serra & Arcos, 2014).

As can be seen from Fig.7 (a), chosen parameter for cDTW is totally different for each dataset that indicates the desirable parameter for cDTW-δ is highly related to dataset characteristics but in most cases the lowest variant (it mean 1%) of cDTW seems to be the proper parameter. From Fig. 7 (b) it can be seen that for WDTW-g the first and second variants are consistent for most of case but the value of g is not limited to the first and second variant because for example for CROSS dataset the fifth variant of WDTW-g is chosen and make the WDTW as the best performer for that dataset. It means the whole selected interval for g is effective and need to be checked for every new MTS dataset. It is seen, for LCSS-ϵ , EDR-ϵ, SWALE-ϵ, SWALE-g_c, and TWED-υ the percentage of chosen parameters is focused on two consecutive variants of parameters. It can be resulted probably with finer step size of search grid, more consistent parameter can be achieved. For the ERP technique the default gap value (zero coordinate) was the most selected parameter. On the other hand, TWED-λ and MSM-c parameters show the spread distributions that means they should be selected precisely to get the desirable effect. From Fig. 7 (k) it can be seen for DDTW-α, as a best measure based on the average rank, the chosen parameter mostly distributed in the first and last variant of parameter interval. It shows that for most of case either DTW or differential-

DTW is more effective, hence the weight of one of them is high and the other is low. It can be concluded that the DDTW-α is also exceedingly reliant to the dataset characteristics.




(c)
LCSS - ε
1
2
3
4
5
100
80
60
40
20
0
ASL-35
VMT
SM
CROSS
I5
I5SIM1
I5SIM2
I5SIM3
LABOMNI
ASL-10
FT
HC-digit
HC
CAL1
CAL2
CAL3
CAL4
CAL5
CAL6
CAL7
CAL8
CAL9
SIGNATURE
ArabianDigit
ASL2Full
CharTraj
JapVowels
robotLP1
robotLP2
robotLP3
robotLP4
robotLP5
Dataset name
(d)
EDR - ε
1
2
3
4
5
100
80
60
40
20
0
Dataset name
(e)
ERP - gap
1
2
3
100
50
0
Dataset name
(f)
SWALE - ε
1
2
3
4
5
100
80
60
40
20
0
Dataset name
(g)
SWALE - g_c
1
2
3
4
5
100
80
60
40
20
0
Dataset name

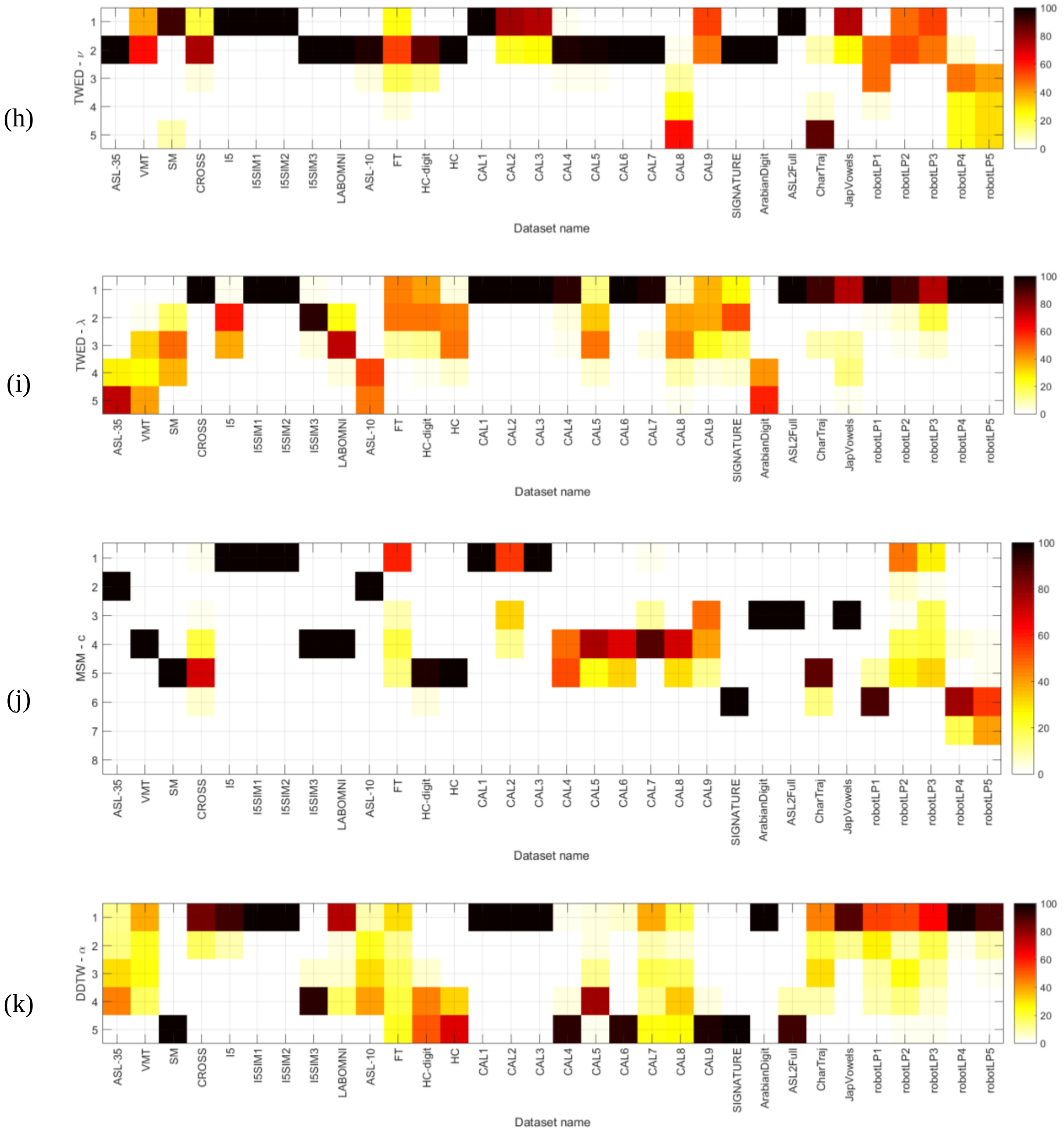


Fig. 7 Percentage of times (color code) that a given parameter value (vertical axis) is chosen for each dataset (horizontal axis). (a) constrained DTW – δ, (b) WDTW – g, (c) LCSS – ϵ, (d) EDR – ϵ, (e) ERP- gap, (f,g) SWALE – ϵ, g_c, (h,i) TWED – ν, λ, (j) MSM- c, (k) DDTW – α.

6. Conclusion

In this paper, 14 similarity measures on 32 publicly available datasets with different characteristics for MTS data were extensively evaluated. The classification performance was evaluated and discussed in detail. Classification performance was assessed in terms of accuracy and statistical significance. Our main findings are as follows:

- The DDTW measure consistently performs better than all considered measures and it was significantly best performing measure in nine MTS datasets, followed by CIDTW measure with performing significantly as the best measure in five datasets.
- The DDTW technique obtained the best average rank among all similarity methods (4.61), followed by WDTW with an average rank of 4.73. Also, SWALE technique showed the worst accuracy in nearly all MTS datasets and showed the weakest average rank (12.77).
- We conclude that there was no specific similarity measure that statistically significantly outperformed the other techniques based on our MTS datasets.
- Finally, based on parameter analysis for most techniques there were some consistent variants of parameter range that could be used in selection of proper parameters for most of similarity measures.

Adding alternative measures and using more MTS datasets may lead to more comprehensive results. Also, the application of similarity measures for a specific goal such as pattern extraction, prediction, vehicle analysis could be investigated.

The large-scale-based experimental evaluation of multiple approaches is necessary for any mature research field, because it opens up your view to select the most appropriate one. Besides getting an idea to use relevant similarity measure it provides the unified framework to compare and analyze MTS data. Adding alternative measures and using more datasets may lead to more comprehensive results. Also, the application of similarity measures for a specific goal, such as pattern extraction, prediction, vehicle analysis could be investigated.

7. Appendixes

7.1. DTW algorithm

Algorithm 2 $D_{DTW}\left(A_1^p, B_1^q\right)$

Input: Two MTS datasets A_1^p and B_1^q

Output: An estimated DTW distance between A_1^p and B_1^q

```
Let $D$ be the $(p+1) \times (q+1)$ initialized to zero
for $i \leftarrow 1p + 1$ do
    $D(i, 1) \leftarrow \infty$
for $j \leftarrow 1q + 1$ do
    $D(1, j) \leftarrow \infty$
for $i \leftarrow 1p$ do
    for $j \leftarrow 1q$ do
        $D(i+1, j+1) \leftarrow D(i, j)$
        if $D(i+1, j+1) > D(i+1, j)$ then
            $D(i+1, j+1) \leftarrow D(i+1, j)$
        if $D(i+1, j+1) >$ then
            $D(i+1, j+1) \leftarrow D(i, j+1)$
        $D(i+1, j+1) \leftarrow D(i+1, j+1) + d_{eucl}\left(a_i, b_j\right)$
return $D(p+1, q+1)$
```

7.2. cDTW algorithm

Algorithm 3 $D_{cDTW}\left(A_1^p, B_1^q, \delta\right)$

Input: Two MTS datasets A_1^p , B_1^q, and windows size δ
Output: An estimated constrained DTW distance between A_1^p and B_1^q

```
Let $D$ be the $(p + 1) \times (q + 1)$ initialized to zero
for $i \leftarrow 1p + 1$ do
    $D(i, 1) \leftarrow \infty$
for $j \leftarrow 1q + 1$ do
    $D(1, j) \leftarrow \infty$
for $i \leftarrow 1p$ do
    for $j \leftarrow 1q$ do
        If $|i - j| \leq \delta$ then
            $D(i + 1, j + 1) \leftarrow D(i, j)$
            if $D(i + 1, j + 1) > D(i + 1, j)$ then
                $D(i + 1, j + 1) \leftarrow D(i + 1, j)$
            if $D(i + 1, j + 1) >$ then
                $D(i + 1, j + 1) \leftarrow D(i, j + 1)$
            $D(i + 1, j + 1) \leftarrow D(i + 1, j + 1) + d_{eucl}(a_i, b_j)$
return $D(p + 1, q + 1)$
```

7.3. WDTW algorithm

Algorithm 4 $D_{WDTW}(A_1^p, B_1^q, g)$

Input: Two MTS datasets A_1^p, B_1^q, and control g
Output: The WDTW distance between A_1^p and B_1^q

```
Let $D$ be the $(p + 1) \times (q + 1)$ initialized to zero
for $i \leftarrow 1p + 1$ do
    $D(i, 1) \leftarrow \infty$
for $j \leftarrow 1q + 1$ do
    $D(1, j) \leftarrow \infty$
for $i \leftarrow 1p$ do
    for $j \leftarrow 1q$ do
        $D(i + 1, j + 1) \leftarrow D(i, j)$
        if $D(i + 1, j + 1) > D(i + 1, j)$ then
            $D(i + 1, j + 1) \leftarrow D(i + 1, j)$
        if $D(i + 1, j + 1) >$ then
            $D(i + 1, j + 1) \leftarrow D(i, j + 1)$
        $w \leftarrow 1/1 + exp\left(-g.\left[|i - j| - \left(\frac{p+q}{2}\right)\right]\right)$
        $D(i + 1, j + 1) \leftarrow D(i + 1, j + 1) + w \times d_{eucl}(a_i, b_j)$
return $D(p + 1, q + 1)$
```

7.4. LCSS algorithm

Algorithm 5 $D_{LCSS}\left(A_1^p, B_1^q, \epsilon\right)$

Input: Two MTS datasets A_1^p, B_1^q and a distance threshold ϵ
Output: The longest common subsequence distance between A_1^p and B_1^q

```
Let $D$ be the $(p+1) \times (q+1)$ initialized to zero
for $i \leftarrow 1p$ do
    for $j \leftarrow 1q$ do
        if $d_{eucl}\left(a_i, b_j\right) < \epsilon$ then
            $D(i+1, j+1) \leftarrow D(i, j) + 1$
        else if $D(i, j+1) > D(i+1, j)$ then
            $D(i+1, j+1) \leftarrow D(i, j+1)$
        else
            $D(i+1, j+1) \leftarrow D(i+1, j)$
return $D(p+1, q+1)$
```

7.5. EDR algorithm

Algorithm 6 $D_{EDR}\left(A_1^p, B_1^q, \epsilon\right)$

Input: Two MTS datasets A_1^p, B_1^q and a distance threshold ϵ
Output: A calculated EDR distance between A_1^p and B_1^q

```
Let $D$ be the $(p+1) \times (q+1)$ initialized to zero
for $i \leftarrow 1p$ do
    $D(i+1,1) \leftarrow i$
for $j \leftarrow 1q$ do
    $D(1, j+1) \leftarrow j$
for $i \leftarrow 1p$ do
    for $j \leftarrow 1q$ do
        if $d_{eucl}\left(a_i, b_j\right) < \epsilon$ then
            $subcost = 0$
        else
            $subcost = 1$
        $D(i+1, j+1) \leftarrow D(i, j) + subcost$
        if $D(i+1, j+1) < D(i, j+1) + 1$ then
            $D(i+1, j+1) = D(i, j+1) + 1$
        else if $D(i+1, j+1) < D(i+1, j) + 1$ then
            $D(i+1, j+1) \leftarrow D(i+1, j) + 1$
return $D(p+1, q+1)$
```

7.6. ERP algorithm

Algorithm 7 $D_{ERP}\left(A_1^p, B_1^q, gap\right)$

Input: Two MTS datasets A_1^p, B_1^q and a sample to calculate gap penalty gap
Output: An evaluated ERP distance between A_1^p and B_1^q

```
Let $D$ be the $(p + 1) \times (q + 1)$ initialized to zero
for $i \leftarrow 1p$ do
    $D(i + 1,1) \leftarrow D(i, 1) + d_{eucl}(a_i, gap)$
for $j \leftarrow 1q$ do
    $D(1, j + 1) \leftarrow D(1, j) + d_{eucl}(gap, b_j)$
for $i \leftarrow 1p$ do
    for $j \leftarrow 1q$ do
        $D(i + 1, j + 1) \leftarrow D(i, j) + d_{eucl}(a_i, b_j)$
        if $D(i + 1, j + 1) > D(i, j + 1) + d_{eucl}(a_i, gap)$ then
            $D(i + 1, j + 1) \leftarrow D(i, j + 1) + d_{eucl}(a_i, gap)$
        if $D(i + 1, j + 1) > D(i + 1, j) + d_{eucl}(gap, b_j)$ then
            $D(i + 1, j + 1) \leftarrow D(i + 1, j) + d_{eucl}(gap, b_j)$
return $D(p + 1, q + 1)$
```

7.7. SWALE algorithm

Algorithm 8 $D_{SWALE}(A_1^p, B_1^q, \epsilon, g_c, r_m)$

Input: Two MTS datasets A_1^p, B_1^q, penalty cost g_c, and reward value r_m
Output: The SWALE distance between A_1^p and B_1^q

```
Let $D$be the $(p + 1) \times (q + 1)$ initialized to zero
for $i \leftarrow 1p$ do
    $D(i + 1,1) \leftarrow g_c \times i$
for $j \leftarrow 1q$ do
    $D(1, j + 1) \leftarrow g_c \times j$
for $i \leftarrow 1p$ do
    for $j \leftarrow 1q$ do
        if $d(a_i, q_j) < \epsilon$ then
            $D(i + 1, j + 1) \leftarrow D(i, j) + r_m$
        else if $D(i + 1, j) > D(i, j + 1)$ then
            $D(i + 1, j + 1) \leftarrow D(i + 1, j) + g_c$
        else
            $D(i + 1, j + 1) \leftarrow D(i, j + 1) + g_c$
return $D(p + 1, q + 1)$
```

7.8. TWED algorithm

Algorithm 9 $D_{TWED}(A_1^p, B_1^q, \nu, \lambda)$

Input: Two MTS datasets A_1^p, B_1^q, stiffness value ν, and the penalty cost λ
Output: The estimated time warp edit distance between A_1^p and B_1^q

```
Let D be the $(p + 1) \times (q + 1)$ initialized to zero
```

```
**for** $i \leftarrow 1p$ **do**
    $D(i+1,1) \leftarrow \infty$
**for** $j \leftarrow 1q$ **do**
    $D(1,j+1) \leftarrow \infty$
**for** $i \leftarrow 1p$ **do**
    **for** $j \leftarrow 1q$ **do**
        **if** $i > 1 \wedge j > 1$ **then**
            $D(i+1,j+1) \leftarrow D(i,j) + 2 \times \nu \times |i-j| + d_{eucl}(a_i, b_j) + d_{eucl}(a_{i-1}, b_{j-1})$
        **else**
            $D(i+1,j+1) \leftarrow D(i,j) + \nu \times |i-j| + d_{eucl}(a_i, b_j)$
        **if** $i > 1$**then**
            $temp \leftarrow D(i,j+1) + \nu + \lambda + d_{eucl}(a_i, a_{i-1})$
        **else**
            $temp \leftarrow D(i,j+1) + \nu + \lambda + d_{eucl}(a_i, 0)$
        **if** $temp < D(i+1,j+1)$ **then**
            $D(i+1,j+1) \leftarrow temp$
        **if** $j > 1$**then**
            $temp \leftarrow D(i+1,j) + \nu + \lambda + d_{eucl}(b_j, b_{j-1})$
        **else**
            $temp \leftarrow D(i+1,j) + \nu + \lambda + d_{eucl}(b_j, 0)$
        **if** $temp < D(i+1,j+1)$ **then**
            $D(i+1,j+1) \leftarrow temp$
**return** $D(p+1, q+1)$
```

7.9. MSM algorithm

Algorithm 10 $MSM(A_1^p, B_1^q, c)$

Input: Two MTS datasets A_1^p, B_1^q, cost value c

Output: The MSM distance between A_1^p and B_1^q

```
Let $D$ be the $p \times q$ initialized to zero
$D(1,1) \leftarrow d_{eucl}(a_1, b_1)$
**for** $i \leftarrow 2p$ **do**
    $D(i,1) \leftarrow D(i-1,1) + cost_{MSM}(a_i, a_{i-1}, b_1, c)$
**for** $j \leftarrow 2q$ **do**
    $D(1,j) \leftarrow D(1,j-1) + cost_{MSM}(b_j, a_1, b_{j-1}, c)$
**for** $i \leftarrow 2p$ **do**
    **for** $j \leftarrow 2q$ **do**
        $D(i,j) \leftarrow D(i-1,j-1) + d_{eucl}(a_i, b_j)$
        **if** $D(i,j) > D(i-1,j) + cost_{MSM}(a_i, a_{i-1}, b_j, c)$ **then**
            $D(i,j) \leftarrow D(i-1,j) + cost_{MSM}(a_i, a_{i-1}, b_j, c)$
        **if** $D(i,j) > D(i,j-1) + cost(b_j, a_i, b_{j-1}, c)$ **then**
            $D(i,j) \leftarrow D(i-1,j) + cost(b_j, a_i, b_{j-1}, c)$
**return** $D(p,q)$
```

7.10. Hausdorff algorithm

Algorithm 11 $D_{Hausdorff}\left(A_1^p, B_1^q\right)$

Input: Two MTS datasets A_1^p and B_1^q
Output: The Hausdorff distance between A_1^p and B_1^q

```
$D_1 \leftarrow 0$
**for** $i \leftarrow 1p$ **do**
    $D_{min1} \leftarrow \infty$
    **for** $j \leftarrow 1q-1$ **do**
        $temp \leftarrow d_{p2s}\left(a_i, bs_j\right)$
        **if** $temp < D_{min1}$ **then**
            $D_{min1} \leftarrow temp$
    **if** $D_1 > D_{min1}$ **then**
        $D_1 \leftarrow D_{min1}$
$D_2 \leftarrow 0$
**for** $i \leftarrow 1q$ **do**
    $D_{min2} \leftarrow \infty$
    **for** $j \leftarrow 1p-1$ **do**
        $temp \leftarrow d_{p2s}\left(b_i, as_j\right)$
        **if** $temp < D_{min2}$ **then**
            $D_{min2} \leftarrow temp$
    **if** $D_2 > D_{min2}$ **then**
        $D_2 \leftarrow D_{min2}$
**if** $D_1 > D_2$ **then**
    $D \leftarrow D_1$
**else**
    $D \leftarrow D_2$
**return** $D$
```

7.11. Discrete Fréchet algorithm

Algorithm 12 $D_{disFrechet}\left(A_1^p, B_1^q\right)$

Input: Two MTS datasets A_1^p and B_1^q
Output: An estimated discrete Fréchet distance between A_1^p and B_1^q

```
Let $D$ be the $p \times q$ matrix initialized to zero
$D(1,1) \leftarrow d_{eucl}(a_1, b_1)$
**for** $i \leftarrow 2p$ **do**
    $D(i,1) \leftarrow d_{eucl}(a_i, b_1)$
    **if** $D(i,1) < D(i-1,1)$ **then**
        $D(i,1) \leftarrow D(i-1,1)$
**for** $j \leftarrow 2q$ **do**
    $D(1,j) \leftarrow d_{eucl}\left(a_1, b_j\right)$
    **if** $D(1,j) < D(1,j-1)$ **then**
        $D(1,j) \leftarrow D(1,j-1)$
```

```
for $i \leftarrow 2p$ do
    for $j \leftarrow 2q$ do
        $D(i,j) \leftarrow D(i-1,j)$
        if $D(i,j) < D(i,j-1)$ then
            $D(i,j) \leftarrow D(i,j-1)$
        if $D(i,j) < D(i-1,j-1)$ then
            $D(i,j) \leftarrow D(i-1,j-1)$
        if $D(i,j) < d_{eucl}(a_i, b_j)$ then
            $D(i,j) \leftarrow d_{eucl}(a_i, b_j)$
return $D(p,q)$
```

7.12. SSPD algorithm

Algorithm 13 $D_{SSPD}(A_1^p, B_1^q)$

Input: Two MTS datasets A_1^p and B_1^q
Output: The SSPD distance between A_1^p and B_1^q

```
$D1 \leftarrow 0$
for $i \leftarrow 1p$ do
    $temp1 \leftarrow \infty$
    for $j \leftarrow 1q-1$ do
        if $d_{p2s}(a_i, bs_j) < temp1$ then
            $temp1 \leftarrow d_{p2s}(a_i, bs_j)$
    $D1 \leftarrow D1 + temp1$
$D2 \leftarrow 0$
for $j \leftarrow 1q$ do
    $temp2 \leftarrow \infty$
    for $i \leftarrow 1p-1$ do
        if $d_{p2s}(b_j, as_i) < temp2$ then
            $temp2 \leftarrow d_{p2s}(b_j, as_i)$
    $D2 \leftarrow D2 + temp2$
return $(D1 + D2)/2$
```

7.13. CIDTW algorithm

Algorithm 14 $D_{CIDTW}(A_1^p, B_1^q)$

Input: Two MTS datasets A_1^p and B_1^q
Output: The evaluated CIDTW distance between A_1^p and B_1^q

```
$comp1 \leftarrow 0$
for $i \leftarrow 1p-1$ do
    $comp1 \leftarrow comp1 + d_{eucl}(a_i, a_{i+1})$
$comp2 \leftarrow 0$
for $j \leftarrow 1q-1$ do
```

```
    $comp2 \leftarrow comp2 + d_{eucl}(b_j, b_{j+1})$
$dist \leftarrow D_{DTW}(A_1^p, B_1^q)$
**if** $comp2 > comp1$ **then**
    $w \leftarrow \frac{comp2}{comp1}$
**else**
    $w \leftarrow \frac{comp1}{comp2}$
**return** $w \times dist$
```

7.14. DDTW algorithm

Algorithm 15 $D_{DDTW}(A_1^p, B_1^q, \alpha)$

Input: Two MTS datasets A_1^p , B_1^q, and combination ratio α
Output: The DDTW distance between A_1^p and B_1^q

```
$dA_1^p \leftarrow diff(A_1^p)$
$dB_1^q \leftarrow diff(B_1^q)$
$dist1 \leftarrow D_{DTW}(A_1^p, B_1^q)$
$dist2 \leftarrow D_{DTW}(dA_1^p, dB_1^q)$
**return** $\alpha \times dist1 + (1 - \alpha) \times dist2$
```

References

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H.-T. (2012). *Learning from data* (Vol. 4): AMLBook New York, NY, USA:.

Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering–A decade review. *Information Systems, 53*, 16-38.

Bagnall, A., Bostrom, A., Large, J., & Lines, J. (2016). The great time series classification bake off: an experimental evaluation of recently proposed algorithms. *Extended Version. CoRR, abs/1602.01711*.

Batista, G. E., Keogh, E. J., Tataw, O. M., & De Souza, V. M. (2014). CID: an efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery*, *28*, 634-669.

Bellman, R. (1956). Dynamic Programming and Lagrange Multipliers. *Proc Natl Acad Sci U S A, 42,* 767-769.

Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop* (Vol. 10, pp. 359-370): Seattle, WA.

Besse, P. C., Guillouet, B., Loubes, J.-M., & Royer, F. (2016). Review and Perspective for Distance-Based Clustering of Vehicle Trajectories. *IEEE Transactions on Intelligent Transportation Systems, 17,* 3306-3317.

Beyan, C., & Fisher, R. B. (2013). Detection of Abnormal Fish Trajectories Using a Clustering Based Hierarchical Classifier. In *BMVC*.

Calderara, S., Prati, A., & Cucchiara, R. (2011). Mixtures of von mises distributions for people trajectory shape analysis. *IEEE Transactions on Circuits and Systems for Video Technology, 21*, 457-471.

Chen, L., & Ng, R. (2004). On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30* (pp. 792-803): VLDB Endowment.

Chen, L., Özsu, M. T., & Oria, V. (2005). Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (pp. 491-502): ACM.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research, 7*, 1-30.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., & Keogh, E. (2008). Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment, 1*, 1542-1552.

Eiter, T., & Mannila, H. (1994). Computing discrete Fréchet distance. In: Citeseer.

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). *Fast subsequence matching in time-series databases* (Vol. 23): ACM.

Fréchet, M. M. (1906). Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940), 22*, 1-72.

Frentzos, E., Gratsias, K., & Theodoridis, Y. (2007). Index-based most similar trajectory search. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on* (pp. 816-825): IEEE.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics, 11*, 86-92.

Fu, T.-c. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence, 24*, 164-181.

García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences, 180*, 2044-2064.

Górecki, T., & Łuczak, M. (2013). Using derivatives in time series classification. *Data Mining and Knowledge Discovery*, 1-22.

Hammami, N., & Bedda, M. (2010). Improved tree model for arabic speech recognition. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on* (Vol. 5, pp. 521-526): IEEE.

Hausdorff, F. (1927). *Mengenlehre*: Walter de Gruyter Berlin.

Hollander, M., Wolfe, D. A., & Chicken, E. (2013). *Nonparametric statistical methods*: John Wiley & Sons.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, 65-70.

Hu, W., Li, X., Tian, G., Maybank, S., & Zhang, Z. (2013). An incremental DPMM-based method for trajectory clustering, modeling, and retrieval. *IEEE Trans Pattern Anal Mach Intell, 35*, 1051-1065.

Jeong, Y.-S., Jeong, M. K., & Omitaomu, O. A. (2011). Weighted dynamic time warping for time series classification. *Pattern recognition, 44*, 2231-2240.

Kadous, W. (1995). GRASP: Recognition of Australian sign language using Instrumented gloves.

Keogh, E. (2011). Machine learning in time series databases (and everything is a time series!). In *Tutorial at the AAAI Int. Conf. on Artificial Intelligence*.

Keogh, E., & Kasetty, S. (2003). On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and Knowledge Discovery, 7*, 349-371.

Keogh, E., Wei, L., Xi, X., Vlachos, M., Lee, S.-H., & Protopapas, P. (2009). Supporting exact indexing of arbitrarily rotated shapes and periodic time series under Euclidean and warping distance measures. *The VLDB Journal—The International Journal on Very Large Data Bases, 18*, 611-630.

Keogh, E. J., & Pazzani, M. J. (2001). Derivative dynamic time warping. In *Proceedings of the 2001 SIAM International Conference on Data Mining* (pp. 1-11): SIAM.

Kleist, C. (2015). *Time Series Data Mining Methods: A Review.* Humboldt-Universität zu Berlin.

Kudo, M., Toyama, J., & Shimbo, M. (1999). Multidimensional curve classification using passing-through regions. *Pattern Recognition Letters, 20*, 1103-1111.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, pp. 707-710).

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition, 38*, 1857-1874.

Lines, J., & Bagnall, A. (2015). Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery, 29*, 565-592.

Lopes, L. S., & Camarinha-Matos, L. M. (1998). Feature transformation strategies for a robot learning problem. In *Feature Extraction, Construction and Selection* (pp. 375-391): Springer.

Marteau, P. F. (2009). Time warp edit distance with stiffness adjustment for time series matching. *IEEE Trans Pattern Anal Mach Intell, 31*, 306-318.

Morris, B., & Trivedi, M. (2009). Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 312-319): IEEE.

Morse, M. D., & Patel, J. M. (2007). An efficient and accurate method for evaluating time series similarity. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (pp. 569-580): ACM.

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

Rabiner, L. R., & Juang, B.-H. (1993). Fundamentals of speech recognition.

Ramos-Garijo, R., Martín, S., Marzal, A., Prat, F., Vilar, J. M., & Llorens, D. (2007). An input panel and recognition engine for on-line handwritten text recognition. *Frontiers in Artificial Intelligence and Applications, 163*, 223.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing, 26*, 43-49.

Salzberg, S. L. (1997). On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery, 1*, 317-328.

Serra, J., & Arcos, J. L. (2014). An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems, 67*, 305-314.

Shokoohi-Yekta, M., Wang, J., & Keogh, E. (2015). On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In *Proceedings of the 2015 SIAM International Conference on Data Mining* (pp. 289-297): SIAM.

Stefan, A., Athitsos, V., & Das, G. (2013). The move-split-merge metric for time series. *IEEE transactions on Knowledge and Data Engineering, 25*, 1425-1438.

Vlachos, M., Kollios, G., & Gunopulos, D. (2002). Discovering similar multidimensional trajectories. In *Data Engineering, 2002. Proceedings. 18th International Conference on* (pp. 673-684): IEEE.

Wagner, R. A., & Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM (JACM), 21*, 168-173.

Wang, H., Su, H., Zheng, K., Sadiq, S., & Zhou, X. (2013). An effectiveness study on trajectory similarity measures. In *Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137* (pp. 13-22): Australian Computer Society, Inc.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., & Keogh, E. (2013). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 1-35.

Wei, W. W.-S. (1994). *Time series analysis*: Addison-Wesley publ Reading.

Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in Statistics* (pp. 196-202): Springer.

Williams, B. H., Toussaint, M., & Storkey, A. J. (2007). A Primitive Based Generative Model to Infer Timing Information in Unpartitioned Handwriting Data. In *IJCAI* (pp. 1119-1124).

Yeung, D.-Y., Chang, H., Xiong, Y., George, S., Kashi, R., Matsumoto, T., & Rigoll, G. (2004). SVC2004: First international signature verification competition. In *Biometric Authentication* (pp. 16-22): Springer.

Zhang, Z., Huang, K., & Tan, T. (2006). Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (Vol. 3, pp. 1135-1138): IEEE.