Locally Slope-based Dynamic Time Warping for Time Series Classification

Anonymous Author(s)

ABSTRACT

Dynamic time warping (DTW) has been widely used in various domains of daily life. Essentially, DTW is a non-linear point-to-point matching method under time consistency constraints to find the optimal path between two temporal sequences. Although DTW achieves a globally optimal solution, it does not naturally capture locally reasonable alignments. Concretely, two points with entirely dissimilar local shape may be aligned. To solve this problem, we propose a novel weighted DTW based on local slope feature (LSDTW), which enhances DTW by taking regional information into consideration. LSDTW is inherently a DTW algorithm. However, it additionally attempts to pair locally similar shapes, and to avoid matching points with distinct neighborhood slopes. Furthermore, when LSDTW is used as a similarity measure in the popular nearest neighbor classifier, it beats other distance-based methods on the vast majority of public datasets, with significantly improved classification accuracies. In addition, case studies establish the interpretability of the proposed method.

KEYWORDS

dynamic time warping, local slope feature, time series alignment, classification

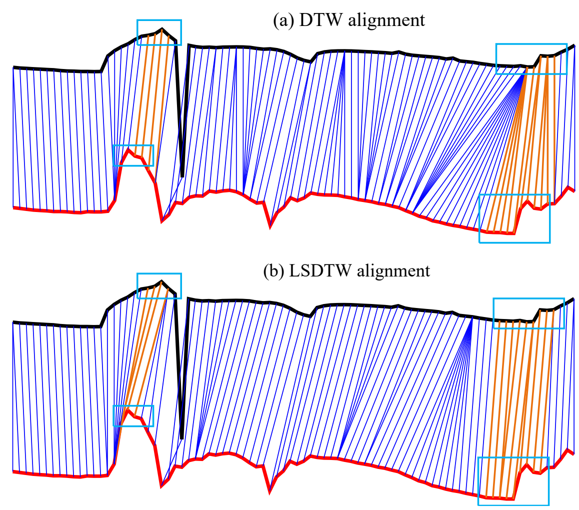

Figure 1: Sequence alignments. (a) DTW alignment. DTW fails to align similar local shapes (e.g. peaks and valleys in the boxed region). (b) LSDTW alignment. We achieve a more accurate and interpretable matching with similarly local structures aligned.

1 INTRODUCTION

Dynamic time warping (DTW) is a dynamic programming algorithm that finds the optimal matching path through the non-linear mapping between sequences or time series. Although it was originally applied to solve the problem of aligning different speed of speech in isolated word recognition [Sakoe and Chiba 2003], researchers have successfully developed it to other fields. For example, handwritten digit recognition [Qiao and Yasuhara 2006], biosignal processing [Boulnemour et al. 2016], etc.

In the field of time series classification, the DTW-based nearest neighbor classifier is one of the benchmark algorithms and is difficult to beat [Bagnall et al. 2017]. It has been successfully applied to classify time series that have global similarity within a class or include a phase shift on the time dimension. DTW is essentially a point-to-point matching method, while additionally enforces time consistencies among aligned point pairs. However, DTW is unreliable and error-prone based only on Euclidean distance, accordingly local different subsequences may be aligned by the DTW (as shown in Fig. 1(a)). This drawback may explain why DTW is less interpretable than shapelet-based classifiers [Hills et al. 2014; Zhao and Itti 2018]. In other words, even though DTW catches the optimal global alignments, it ignores local shape information, which may lead to alignments with little semantic meaning.

In order to address this problem, we propose a **L**ocally **S**lope-based **D**ynamic **T**ime **W**arping algorithm (LSDTW), it not only considers the point-to-point distance, but also takes the local slope similarity between point pairs into account. As a result, we obtain perceptually accurate and interpretable alignments. An alignment example using LSDTW is shown in Fig. 1(b), it is clear that local similarities are matched successfully. Specifically, the more similar the local shape is, the closer the global distance becomes.

Our method is inspired by the filtering techniques and local binary pattern (LBP) [Ojala et al. 2002] that are widely used in the computer vision community. It is well known that the acquisition (even the classification) process of time series is susceptible to noise. Therefore, some filtering techniques are employed to reduce the impact of noise at first. Then we propose a LBP-like one-dimensional time series local shape feature extraction method, which performs local shape coding for each point in the sequence based on slopes of adjacent points. Fig. 2 shows the effectiveness of our proposed method on both global and local similarities.

Our contributions can be summarized as follows: (1) The filtering technique in image processing is applied to time series, which can effectively reduce noises and rectify sequence shape. (2) A novel local slope representation method is proposed to represent and discover local shape features. (3) The proposed LSDTW method is more interpretable, and can significantly improve alignment and classification accuracy.

The rest of our paper is organized as follows. Section 2 describes the related work of this paper. In Section 3, we present the local slope feature coding method. Moreover, the detailed description of LSDTW is introduced in Section 4. Section 5 evaluates the proposed algorithm. Finally, Section 6 provides the conclusion.

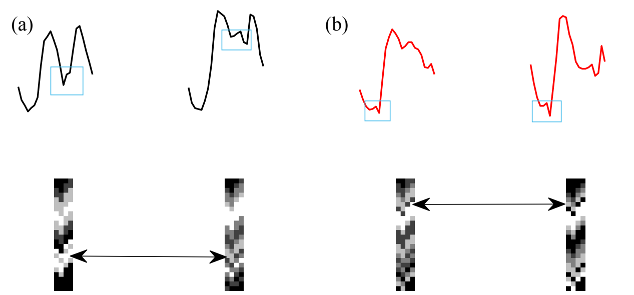

Figure 2: Time series local slope feature coding. (a) The top two sequences are similar globally, while with shape inconsistencies in the boxed area. This global similarity and regional difference are reflected in the following local slope feature coding matrices. (b) The above two sequences from same class have local consistency in the boxed area, which is also reflected in the regional slope feature coding matrix below.

2 RELATED WORK

DTW is a basic sequence alignment method, there exists several ways to improve it. Traditionally, we can add constraints to prevent the matching path from being too distorted, such as Sakoe-Chiba band [Sakoe and Chiba 2003] and the Itakura Parallelogram [Itakura 1975]. Meanwhile, some researchers try to increase the quality of alignment by including weights or other information. In [Jeong et al. 2011], a globally weighted dynamic time warping (WDTW) algorithm that assigns lower weights to points closer to the diagonal is introduced. A locally weighted model that enlarges the margin of *k*NN is presented in [Yuan et al. 2018]. In [Keogh and Pazzani 2001], derivative dynamic time warping (DDTW), which converts the original sequence into a first-order derivatives sequence is proposed to achieve a reasonable alignment. A complexity-invariant DTW by multiplying a complex correction factor is proposed in [Batista et al. 2014]. However, these algorithms neglect local neighborhoods information.

For local shape representation, slope that linearly fits subsequence with a straight line is depicted in [Deng et al. 2013] and [Baydogan et al. 2013]. Histogram of oriented gradients is introduced in [Zhao and Itti 2016] to represent 1D time series sequences. To achieve a reasonable match of local shape features, Zhao *et al.* [Zhao and Itti 2018] converted the sequence into shape feature codes and then aligned them. Although these methods produce relatively better representations or alignments, they are either sensitive to noise or ignore the original global numerical values of time series. On the other hand, our method not only considers local and global information, but also employs filtering and discretization techniques to effectively improve the alignment quality.

Based on the classification strategy, the time series classifier can be divided into two types, distance-based, and feature-based. The first one combines different distance measures with nearest neighbor classifiers. Apart from the methods described above for sequence alignment, researchers also proposed other measures. For example, time warping edit (TWE) [Marteau 2009], move-split-merge (MSM) [Stefan et al. 2013] and so on. The second one transforms the original sequence into a new feature space, which can effectively extract discriminate features. The representative methods include the time series forest (TSF) model proposed in [Baydogan et al. 2013] and [Shi et al. 2018], learned patterns similarity (LPS) that uses nonlinear regression relationships between segments of the time series [Baydogan and Runger 2016], sequence learning based on symbolic aggregate approximation (SAX) [Nguyen et al. 2017, 2018] and bag-of-pattern model based on Fourier coefficients [Schäfer 2015; Schäfer and Leser 2017].

Another category of feature-based method depends mostly on the uninterpretable convolutional neural networks (CNN) to extract complex features for classification. For example, Time LeNet (t-LeNet) was explored at first by Le Guennec *et al.* [Le Guennec et al. 2016] to extend CNN to time series classification problems. Cui *et al.* [Cui et al. 2016] proposed a multi-scale CNN method (MCNN), it trains CNN based on extracted subsequences. In addition, Multi Layer Perceptron (MLP), Fully Convolutional Neural Network (FCN), and Residual Network (ResNet) are successfully applied on time series [Wang et al. 2017]. Furthermore, Zhao *et al.* [Zhao et al. 2017] proposed the Time-CNN algorithm, it replaces the cross entropy loss function with mean squared error for adapting time series. For other neural networks, Tanisaro *et al.* [Tanisaro and Heidemann 2017] proposed a variant of Echo State Networks that directly uses the original time series to predict the probability distribution of the class variables. SerrÃă *et al.* [SerrÃă et al. 2018] proposed an encoder algorithm similar to FCN, which is combined with the attention layer and the PReLU activation function instead. Unlike the above algorithms, our single model can also be applied for accurate sequence alignment, which is faster, more interpretable and practical.

3 LOCAL SLOPE FEATURE CODING

This section presents the steps needed to represent local shape features. A filtering method is considered at first for noise reduction. Then slope values discretization and encoding algorithms are explained.

3.1 Filtering

There are various types of filters in image processing, including mean filtering, Gaussian filtering etc. For simplicity, we select the simple mean filtering to reduce noise. It essentially replaces a point in the sequence with mean value of adjacent ones. Given a time series $T = \{t_1, t_2, ..., t_n\}$, assuming the length of filtering window is w, and the filtered sequence is $T^f = \{t_1^f, t_2^f, ..., t_n^f\}$, where

$$t_i^f = \frac{1}{w} \sum_{j=i-w/2}^{i+w/2} t_i, \qquad w/2 < i < n - w/2. \tag{1}$$

As can be seen from Fig. 3, filtering can effectively weaken local fluctuations of sequence caused by noise, and rectify the global shape of it. In the subsequent experiments, we will further analyze the selection of w in our model.

3.2 Discretization

In this paper, we use the slope of time series to represent local characteristics of sequence. Unlike the slopes used in previous

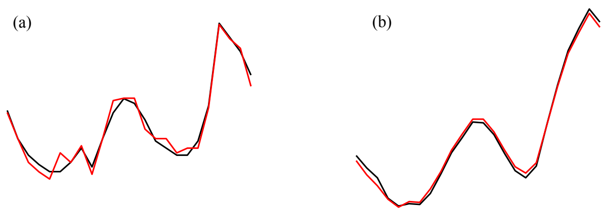

Figure 3: Two sequences of the same type in *ItalyPowerDemand* dataset. (a) The original sequences. This two sequences have obvious differences in shape due to noise. (b) The filtered sequences. The two sequences are almost identical after mean filtering.

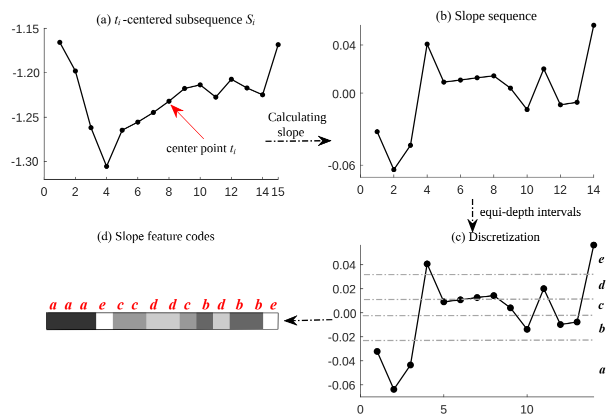

Figure 4: Local slope feature coding process of point t_i.

works [Baydogan et al. 2013; Deng et al. 2013], we only measure the slope between adjacent points without linearly fitting subsequences. The slope of two points is calculated as:

$$s_i = \frac{t_{i+1} - t_i}{\Delta t}, \qquad 1 \le i < n, \tag{2}$$

where Δt represents the time interval of these two points. Normally, we consider the time interval between adjacent points as an unit time, that is $\Delta t = 1$. The top two diagrams of Fig.4 depict the process of slope representation.

For denoising of slopes, we try to discretize the obtained continuous slopes to symbolic ones. According to the division strategy, it can be divided into three categories, namely *equi-depth*, *equi-width* and *optimization*. The first two methods are unsupervised, while the last one supervised. As shown in the Fig. 4(c), the *equi-depth* is adopted. Given a continuous data sequence $T = \{t_1, t_2, ..., t_n\}$, we divide these points into k discrete values $\{p_0, p_1, ..., p_k\}$, where p_0 and p_k are the minimum and maximum values respectively. For *equi-width*, the length of each interval $[p_i, p_{i+1})$ is equal, while the number of data points in each interval $[p_i, p_{i+1})$ is equal for the *equi-depth*.

3.3 Feature Coding

Given a filtered time series of length n, $T = \{t_1, t_2, ..., t_n\}$. First, we extract subsequences S of length l from each temporal point t. The subsequence S_i is centred on t_i(as shown in the Fig. 4(a)). The length l defines the number of neighborhoods around t_i. When $l = 1$, no local neighborhood information is considered. With the increase of l, more neighborhoods are included, and in the extreme case when $l = n$, subsequences sampled from different points become the same. Second, the subsequence S is converted into a slope sequence of length l-1 (as shown in the Fig. 4(b)). Finally, each slope sequence is converted to local slope feature coding symbols using the *equi-depth* method, and represented by a grey scale map visually (as shown in the Fig. 4(c-d)). Since each point t_i is transformed to local slope feature codes of length l-1, the whole sequence T will be represented by a local slope feature coding matrix of n*(l-1) symbols. Fig. 5 exhibits the denoising ability of *equi-depth* discretization, illustrating that the proposed local coding method is powerful and interpretable.

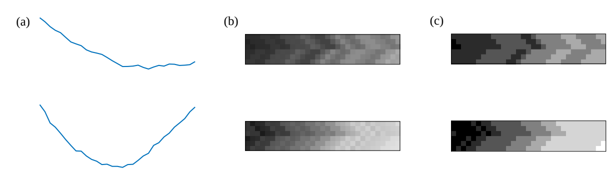

Figure 5: Effectiveness of discretized slope coding matrix. (a) Two subsequences have big differences in the second half. (b) Shape features have no visible boundaries without discretization, and differences between two coding matrices are not significant. (c) Clear boundaries appear after discretization, the former part between two matrices are similar while the latter part distinctive.

4 THE PROPOSED LSDTW APPROACH

In this section, we will briefly introduce the dynamic time warping at first. Then the proposed LSDTW based on local shape features is discussed in detail.

4.1 Dynamic Time Warping

DTW finds the optimal matching path between sequences by nonlinear mapping. It can be applied to both univariate and multivariate time series. For simplicity, we only consider the univariate time series with equal length in this paper.

Given two time series $T^A = \{t_1^A, t_2^A, ..., t_n^A\}$ and $T^B = \{t_1^B, t_2^B, ..., t_n^B\}$ of length n. We define the matrix $\mathbf{C} \in \mathcal{R}^{n \times n}$, where each element $c(i, j)$, $1 \le i, j \le n$ represents the shortest distance between the subsequence $T^{A'} = \{t_1^A, t_2^A, ..., t_i^A\}$ and $T^{B'} = \{t_1^B, t_2^B, ..., t_j^B\}$. A warping path $P = ((e_1, f_1), (e_2, f_2), ..., (e_q, f_q))$, $n \le q \le 2n - 1$ is a series of points (*i.e.* pairs of indexes) that define a traversal of matrix $\mathbf{C}$. A valid warping path must satisfy two conditions:

- Endpoint constraint: the start and end points of the matching path must be the first and last point pairs of the matrix $\mathbf{C}$, *i.e.* $(e_1, f_1) = (1, 1)$ and $(e_q, f_q) = (n, n)$.
- Monotonic continuity constraint: for any $i < n$, it has $0 \le e_{i+1} - e_i \le 1$ and $0 \le f_{i+1} - f_i \le 1$.

The optimal matching between the two sequences is achieved by minimizing the accumulated distance. The recursive algorithm is given below.

Algorithm 1 LSDTW Distance

Input: two time series T^A, T^B
k, alphabet size for slope discretization,
w, the filter window size,
l, the length of subsequence,
α controls the influence of local shape features
Output: $LSDTW(T^A, T^B)$

1: Filter the sequences T^A and T^B based on Eq. (1)
2: Initialize slope feature coding matrices $\mathbf{M}^A$ and $\mathbf{M}^B$
3: **for** each t_i^A, t_j^B, $1 \le i, j \le n$ **do**
4: Extract centered subsequences $T_i^{A'}$, $T_j^{B'}$ with length l
5: Calculate the slopes of $T_i^{A'}$, $T_i^{B'}$ based on Eq. (2) and discretize them to form local feature vectors M_i^A, M_j^B
6: **end for**
7: Calculate $LSDTW(T^A, T^B)$ recursively using Eq. (5) - (8)
8: **return** LSDTW Distance $LSDTW(T^A, T^B)$

$$c(i,j) = |t_i^A - t_j^B|^2 + min\{c(i-1,j-1), c(i-1,j), c(i,j-1)\} \quad (3)$$

$$DTW(T^A, T^B) = c(n,n) \quad (4)$$

4.2 Locally Slope-based DTW

DTW finds an optimal matching path globally under certain constraints, but it is difficult to achieve a reasonable matching of local shapes. For that, we combine the local shape similarity measure of point pairs with the DTW matching process to attain an interpretable and reasonable alignment. In our LSDTW, the more similar the local shape is, the closer the point-to-point function is, indeed the easier it is to match them successfully.

Given two time series $T^A, T^B \in \mathcal{R}^n$. The generated local slope feature coding matrices are $\mathbf{M}^A = \{M_1^A, M_2^A, \ldots, M_n^A\}$ and $\mathbf{M}^B = \{M_1^B, M_2^B, \ldots, M_n^B\}$, $M_i^A, M_j^B \in \mathcal{R}^{l-1}$, $\mathbf{M}^A, \mathbf{M}^B \in \mathcal{R}^{n\times(l-1)}$. Then we use the DTW framework for sequence alignment.

$$d_{ij} = \alpha Sim(M_i^A, M_j^B) + (1-\alpha)|t_i^A - t_j^B|^2 \quad (5)$$

$$c(i,j) = d_{ij} + min\{c(i-1,j-1), c(i-1,j), c(i,j-1)\} \quad (6)$$

$$LSDTW(T^A, T^B) = c(n,n) \quad (7)$$

As shown in Eq. (8), $Sim(M_i^A, M_i^B)$ defines the local shape similarity measure of t_i^A and t_j^B.

$$Sim(M_i^A, M_j^B) = \sum_{g=1}^{l-1} |m_{ig}^A - m_{jg}^B| \quad (8)$$

For Eq. (5), $0 \le \alpha \le 1$ is a balance factor, it represents the importance of local slope features during sequence alignment. When there are obvious similarity of local slope features, α takes a relatively larger value, or it accepts a smaller one. If $\alpha = 0$ or $Sim(M_i^A, M_j^B) = 0$, LSDTW degenerates into standard DTW. Therefore, LSDTW is a generalization of DTW. For the calculation of Eq. (8), a very simple one as $|a-b| = 1$, $|c-a| = 2$ is adopted.

Algorithm 1 details the learning process. It is clear that the main differences between DTW and LSDTW lie in the filtering (line 1) and local slope feature coding steps (lines 3-6). Note that the time complexity of DTW is $O(n^2)$, our LSDTW has an additional local shape similarity measure compared with DTW, and its time complexity is $O(ln^2)$. Therefore, the time complexity of LSDTW is $O(n^2 + ln^2)$. Since l is a constant value, and $l \ll n$. LSDTW's complexity can be deduced as $O(n^2 + ln^2) \approx O(n^2)$.

5 EXPERIMENT AND EVALUATION

In this section, experiments are conducted on 85 public datasets[1]. We first describe the parameter setting that effects the experiments, then present the comparisons with advanced classifiers, before discussing the interpretability of the proposed locally aware measure and impact of design decisions.

5.1 Parameter Analysis

As shown in Algorithm 1, there are 4 parameters for us to tune. In this section, the range of values considered for these parameters are as follows. The alphabet size $k \in [3,8]$, the length of subsequence $l \in [3,21]$, the balance factor $\alpha \in [0:0.02:1]$, and the filter window size is the same as l. When one of the parameters varies, a basis configuration of other three is selected from $\alpha = 0.02$, l=11, w=5, and k=7 for simplicity.

Fig. 6 shows the trends of classification accuracy when different parameters vary. First, the accuracies of most datasets rise with the increase of l, and tend to be stable when l reaches a certain value (e.g. $l = 11$). Second, for the filtering window w, except for *SonyAIBORobotS*1 and *MiddlePhalanxOA*, the accuracy curves decrease while w increases. Since a larger filter window will loss local detail information, resulting in reduced discriminability between sequences. Therefore, a relative small value (e.g. $w = 5$) will be a better choice. Third, for the balance factor α, on account of the Euclidean distance between two points is usually much smaller than its slope similarity, e.g. $|t_i^A - t_j^B|^2 \ll Sim(M_i^A, M_j^B)$, a smaller value should be consider to avoid that local slope features dominate the distance measure. The down-left sub-figure coincides this idea. It is easy to find that when α reaches a certain value (e.g. $\alpha = 0.4$), the accuracies tend to be stable. However, a great fluctuation emerges when α is relatively small. For example, the differences between the maximum and minimum accuracies of the *DistalPhalanxOA* and *MiddlePhalanxOA* are 20.69% and 24.64%, respectively, and the average margin on the 7 datasets is also as high as 12.32%. Finally, it is clear that an increasing or steady tendency appears with the increment of k (except for *DistalPhalanxOA*).

In a nutshell, for fairness and simplicity, $l = 11$, $w = 5$ and $k = 7$ are considered, while α is achieved by cross-validation in range [0, 0.4] to shorten training time, and to reach better performance.

5.2 Classification

In this section, we classify time series by combining LSDTW with 1NN, and compare its accuracy with other benchmark classifiers.

5.2.1 Compared with Distance-based Classifiers. As a novel time series elastic distance measure, LSDTW is compared with several popular distance-based methods. For example, classic ED and DTW, DDTW [Keogh and Pazzani 2001], WDTW [Jeong et al. 2011], complexity-invariant DTW (CID) [Batista et al. 2014], locally

[1] All the datasets are avaliable at http://www.cs.ucr.edu/~eamonn/time_series_data/

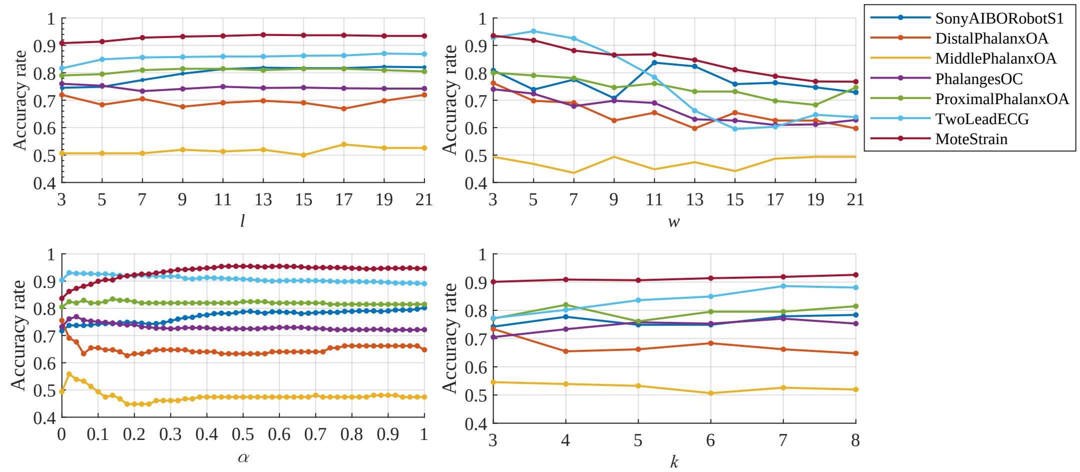

Figure 6: Accuracy rate under different parameters.

weighted DTW (LWDTW) [Yuan et al. 2018], shapeDTW [Zhao and Itti 2018], MSM [Stefan et al. 2013], TWE [Marteau 2009], and longest common subsequence (LCSS). Fig. 7 shows the accuracy comparison between LSDTW and other nine representative alternatives [2]. Each point in the graph represents a dataset, and falling in the lower triangle area indicates that LSDTW works better.

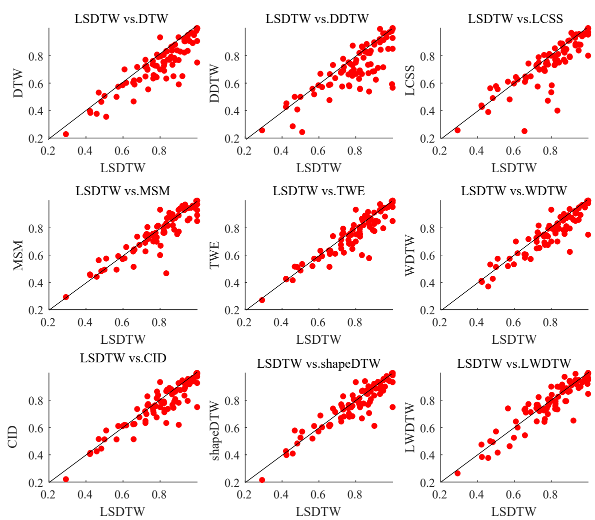

Figure 7: Accuracy comparison between LSDTW and different distance-based classifiers on 85 datasets.

Fig. 7 demonstrates that LSDTW outperforms the existing distance-based approaches on most datasets. E.g. LSDTW is superior (including equal) to DTW on 75 out of 85 datasets, and the accuracy is improved by more than 10% on 17 datasets. Especially, the accuracy of *OSULeaf*, *CinCECGtorso* and *BirdChicken* is increased by 30.99%,

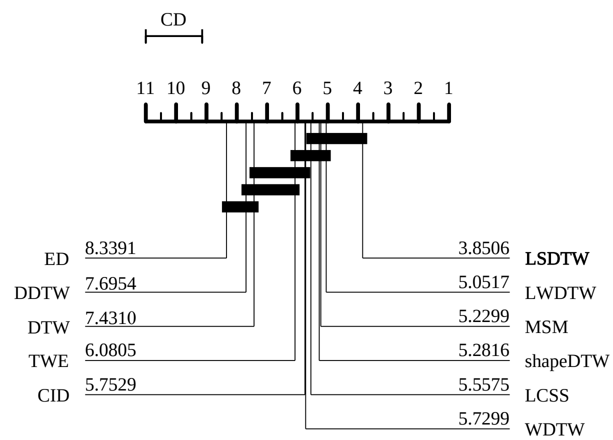

Figure 8: CD diagram for distance-based classifiers.

26.96% and 25.00% respectively. Compared with DDTW, WDTW, CID, LWDTW and shapeDTW, which are also variants of DTW, LSDTW performs better on 71, 65, 62, 56 and 52 datasets respectively. In addition, LSDTW beats LCSS, MSM, and TWE on 58, 56, and 63 datasets (including draws). A more detailed version for the number of Win/Draw/Loss between LSDTW and other compared 37 classifiers is illustrated in Table 1.

Fig. 8 shows a critical difference (CD) diagram [Benavoli et al. 2017; Demsar 2006] over the average ranks of the tested classification methods. Classifier with the lowest (best) rank lies in the upper right corner. The group of classifiers that are not significantly different is connected by a bar. Although it is not easy to say that there is significant difference among these approaches, LSDTW is definitely the best distance-based single classifier as far as we know.

5.2.2 Compared with Feature-based Classifiers. In this section we try to analyse LSDTW with state-of-the-art feature-based ones

[2] The source code of LSDTW and more experimental results could be found on our supporting website https://sites.google.com/view/lsdtw/

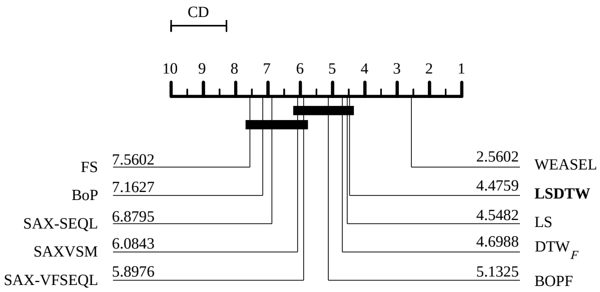

Figure 9: CD diagram for feature-based classifiers.

like fast shapelet (FS) [Rakthanmanon and Keogh 2013], learning shapelet (LS) [Grabocka et al. 2014], DTW features (DTW$_F$) [Kate 2016], Bag of Patterns (BOP) [Lin et al. 2012], Bag-of-Pattern-Features (BOPF) [Li and Lin 2017], Symbolic Aggregate approXimation Vector Space Model (SAXVSM) [Senin and Malinchik 2013], symbolic sequence classifiers like SAX-SEQL and SAX-VFSEQL [Nguyen et al. 2017, 2018] and word extraction for time series classification (WEASEL)[Schäfer and Leser 2017]. From the Fig. 9, LSDTW again is the first except WEASEL. Compared with feature-based classifiers, the main disadvantage of distance-based ones is its less interpretability. However, by considering the local slope information, our LSDTW is more accurate, and interpretable, details will be discussed in Section 5.5. In addition, as an alignment or distance measure, LSDTW may have wider application prospects.

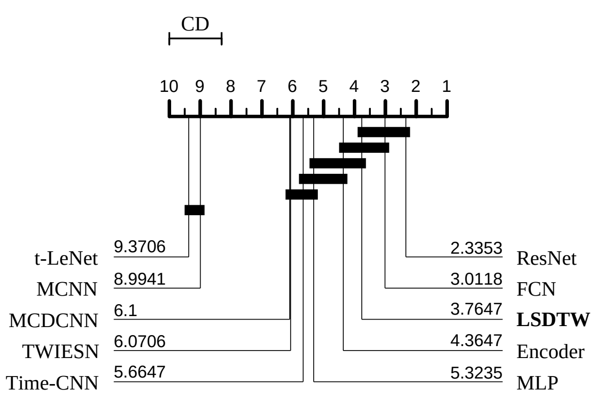

Figure 10: CD diagram for deep learning classifiers.

5.2.3 Compared with Deep Learning Classifiers. Deep learning is also widely used in the field of time series classification. We also tried to compare LSDTW with the nine deep learning methods mentioned in [Fawaz et al. 2018]. They are MLP, FCN, ResNet [Wang et al. 2017], Encoder [SerrÃă et al. 2018], MCNN [Cui et al. 2016], t-LeNet [Le Guennec et al. 2016], Multi Channel Deep Convolutional Neural Network (MCDCNN) [Zheng et al. 2014] , Time-CNN [Zhao et al. 2017], and Time Warping Invariant Echo State Network (TWIESN) [Tanisaro and Heidemann 2017] .Although our method is slightly worse than ResNet and FCN (as shown in Fig. 10), there is not significant difference among them. LSDTW is more interpretable and faster than deep learning methods. In addition, LSDTW can also be used for time series alignment.

5.2.4 Compared with Ensemble Classifiers. Ensemble classifier can achieve good performance through integrating a series of weak classifiers. For time series classification, famous ones include time series forest (TSF) [Deng et al. 2013], time series bag of features (TSBF) [Baydogan et al. 2013], shapelet transform (ST) [Hills et al. 2014], learning pattern similarity (LPS) [Baydogan and Runger 2016], BOSS [Schäfer 2015], elastic ensemble (EE) and the collective of transformation-based ensembles (COTE) [Bagnall et al. 2015], etc. Although our method is slightly worse than COTE, ST and BOSS (as shown in the Fig. 11), there is not significant difference among them. Furthermore, LSDTW is just a single classifier, it is surely faster than these complex method which may combined more than 50 (or even 500) single different algorithms. It also outperforms some notable ones like EE, TSF, TSBF, etc on more than one half of the tested datasets. Furthermore, as a single classifier, LSDTW could be easily combined with ensemble ones like EE, ST and COTE to improve accuracy.

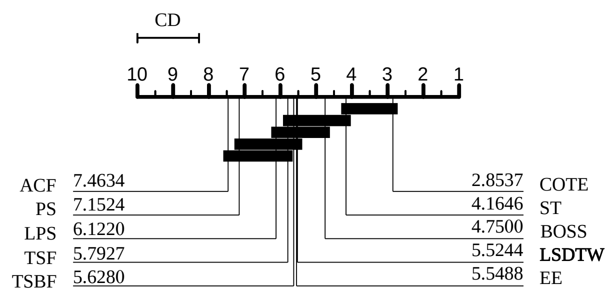

Figure 11: CD diagram for ensemble classifiers.

5.3 Running time

This section will demonstrate that although LSDTW may worse than ensemble classifiers in accuracy, it is more efficient than them. The running time of the memory-consuming WEASEL is not given due to the source code provided is parallel. However, its time complexity is O(Nn^3logn), which is obviously larger than LSDTW's (where N is the number of instances in training set). Since ResNet and FCN with thousands of parameters to tune run on GPU, their running time are not listed for fairness. Fig. 12 shows the 10 times average running time (in log) of BOSS, COTE, ST and LSDTW on 21 relatively small datasets. It is easy to find that LSDTW is definitely the most efficient one, with at least one order of magnitude improving compared with COTE and ST.

5.4 Impact of Design Decisions

The LSDTW model is based on two design decisions.

- Discretization using *equi-depth* as opposed to *equi-width* or *optimization.*
- Filtering as a parameter in contrast to non-filtering.

In this experiment, 1NN is chose as it does not introduce any new parameters, and allows us to focus on the model itself. The scatter plots in Fig. 13 justify the effectiveness of each design decision.

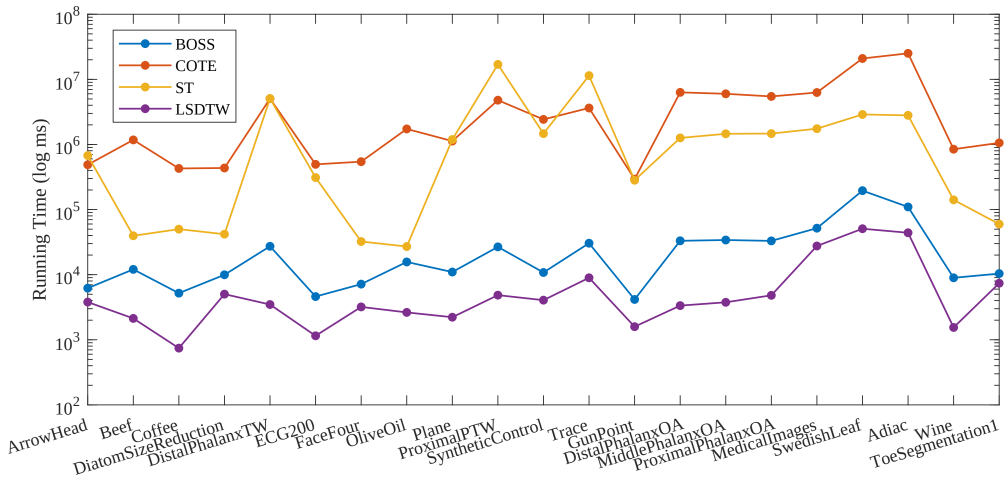

Figure 12: The average running time of BOSS, COTE, ST and LSDTW on 21 datasets.

Table 1: LSDTW versus the other 37 classifiers on 85 datasets. Win/Draw/Loss indicate that LSDTW has Higher/Equal/Lower accuracy than the compared method.

	Win/Draw/Loss		Win/Draw/Loss
ED	68/1/16	BOPF	48/3/32
DTW	62/13/10	SAX-VFSEQL	53/1/29 *
DDTW	67/4/14	SAX-VESQL	56/3/24 *
LCSS	55/3/27	DTW_F	45/2/38
MSM	55/1/29	LS	42/5/38
TWE	61/2/22	FS	67/2/16
WDTW	61/4/20	BoP	66/1/18
CID	60/2/23	SAXVSM	52/3/30
shapeDTW	49/3/33	WEASEL	26/5/54
LWDTW	53/3/29		
BOSS	31/4/50	MLP	61/0/24
TSF	43/1/41	FCN	27/5/53
TSBF	47/3/35	ResNet	26/4/55
LPS	46/5/34	Encoder	56/0/29
ACF	55/2/28	MCNN	83/0/2
PS	51/2/32	t-LetNet	82/0/3
EE	40/3/42	MCDCNN	61/0/24
COTE	19/7/59	Time-CNN	60/0/25
ST	30/3/52	TWIESN	69/1/15

* 83 datasets are compared with SAX-VFSEQL and SAX-VESQL according to the published results.

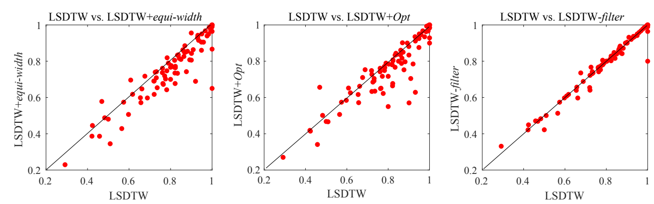

Figure 13: Accuracy comparison under various decisions.

Overall the *equi-depth* shows a better or equal accuracy on 71 of 85 datasets when compared to both *equi-width* and *optimization*. As for filtering, it is clear that filtering-selectively (LSDTW) outperforms its opponents (LSDTW-*filter*), the reason that some points appear in the upper diagonal area may lie in that filtering may decrease the discrimination of some datasets.

5.5 Case Studies

This section will demonstrate that our local slope-based model could achieve better alignment, and is more interpretable when applied to real datasets.

The *BirdChicken* dataset was introduced in [Hills et al. 2014], and its main task is to distinguish between the bird and chicken by the contour of their images. At present, the best result published on this dataset using distance-based method is 95% (attained by shapeDTW [Zhao and Itti 2018]). However, the accuracy of LSDTW can reach 100%. The Fig. 14 shows the sequence alignments of two instances in the *BirdChicken* using DTW, DDTW, shapeDTW and LSDTW respectively. Although both sequences belong to *Chicken* class, DTW, DDTW and shapeDTW misclassified them. Our LSDTW not only successfully classified them as the same class, also aligned the local shapes well (as shown in Fig. 14, blue lines indicate the point-to-point alignments, and the red ones emphasize the warping path for the boxed local sequences).

The *OSULeaf* dataset consists of one-dimensional outlines of leaves. The task on this dataset is to classify the leaves by extracting the boundaries. The *OSULeaf* dataset is rich in local shape features, including many peaks and valleys. We also include DTW, DDTW, shapeDTW and LSDTW for sequence alignment comparisons. As can be seen from Fig. 15, DTW and shapeDTW cannot effectively align the boxed areas. Although DDTW aligns them globally, it fails to align the peaks and valleys inside. Our LSDTW matches the global and local similarities successfully in these areas.

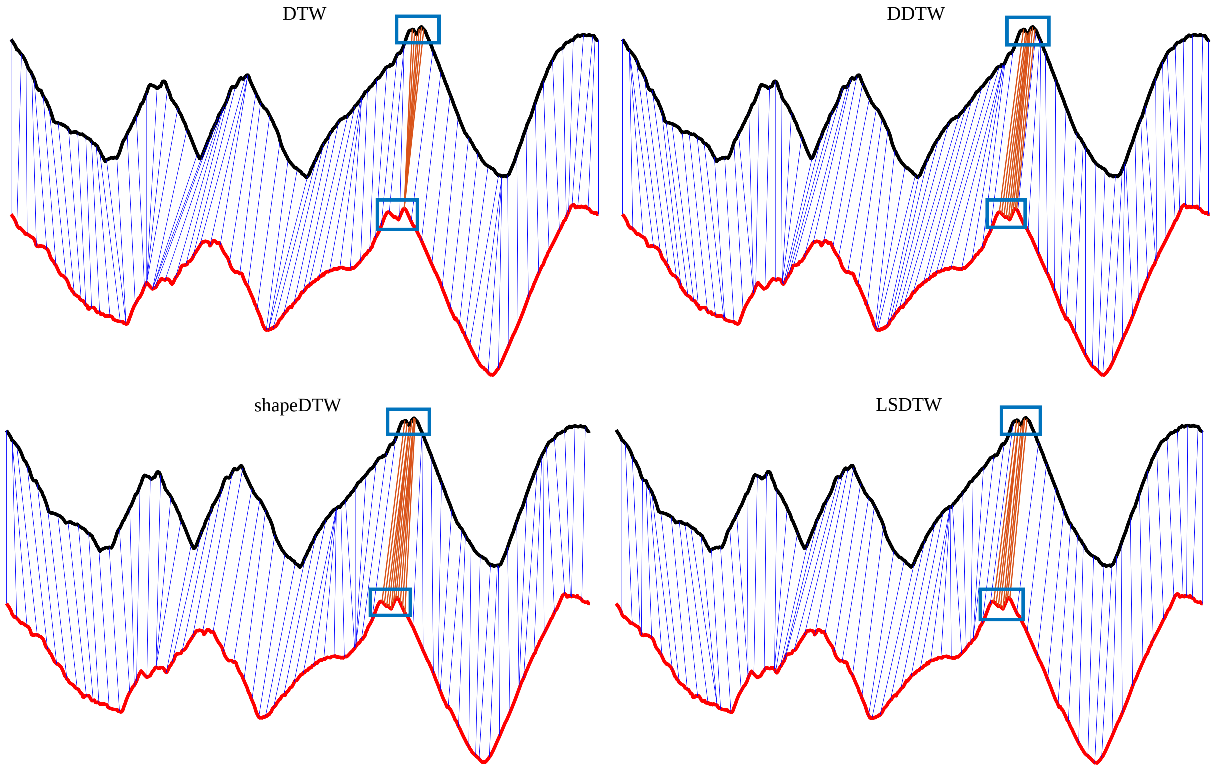

Figure 14: Sequence alignments of DTW, DDTW, shapeDTW and LSDTW on *BirdChicken*. The two subsequences in the box region have similar local shape features (i.e. two peaks and one valley). The DTW, DDTW and shapeDTW cannot align them correctly, while our LSDTW captures these local features well.

In order to further establish the advantages of our interpretable method, t-SNE is employed to discover the underlying structures of the proposed metric learning algorithm [Maaten and Hinton 2008]. Four datasets with the number of class labels varies from 2 to 12 are applied. For example, dataset *MoteStrain* is composed of 2 classes, *Car* of 4, *SyntheticControl* of 6 and *CricketY* of 12. As shown in Fig. 16, it is clear that our LSDTW is the most distinguish one with the largest margins among different classes. For dataset *MoteStrain*, there is an obvious separating line exists between two classes using LSDTW, while other three compared methods mix them. For complex multi-class classification problems like *Car* and *CricketY*, LSDTW isolates parts of classes from others successfully. Although it seems that shapeDTW shows the best result on *SyntheticControl*, it fails to join the class marked with yellow dot together, while our method not only classifies each class correctly, it also acquires a larger margin compared with DTW.

6 CONCLUSION

This paper introduces a DTW model based on local slope features. It represents local shape information visually and clearly, and aligns analogous subsequences effectively by combining coded features with traditional DTW. Experiments on multiple classifiers and various datasets show that our method is more competitive than other distance-based or feature-based classifiers. Besides, case studies illustrate the interpretablility of LSDTW. Future direction of our work could involve investigating how to classify instances with less samples, and apply it directly on other real life problems.

REFERENCES

Anthony Bagnall, Jason Lines, Jon Hills, and Aaron Bostrom. 2015. Time-series classification with COTE: the collective of transformation-based ensembles. *IEEE Trans. Knowl. Data Eng.* 27, 9 (2015), 2522–2535.

Anthony J. Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn J. Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.* 31, 3 (2017), 606–660. https://doi.org/10.1007/s10618-016-0483-9

Gustavo E. A. P. A. Batista, Eamonn J. Keogh, Oben Moses Tataw, and Vinícius M. A. de Souza. 2014. CID: an efficient complexity-invariant distance for time series. *Data Min. Knowl. Discov.* 28, 3 (2014), 634–669. https://doi.org/10.1007/s10618-013-0312-3

Mustafa Gokce Baydogan, George Runger, and Eugene Tuv. 2013. A bag-of-features framework to classify time series. *IEEE transactions on pattern analysis and machine intelligence* 35, 11 (2013), 2796–2802.

Mustafa Gokce Baydogan and George C. Runger. 2016. Time series representation and similarity based on local autopatterns. *Data Min. Knowl. Discov.* 30, 2 (2016), 476–509. https://doi.org/10.1007/s10618-015-0425-y

Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. 2017. Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. *The Journal of Machine Learning Research* 18, 1 (2017), 2653–2688.

Imen Boulnemour, Bachir Boucheham, and Slimane Benloucif. 2016. Improved Dynamic Time Warping for Abnormality Detection in ECG Time Series. In *IWBBIO 2016*. 242–253. https://doi.org/10.1007/978-3-319-31744-1_22

Zhicheng Cui, Wenlin Chen, and Yixin Chen. 2016. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995* (2016).

Janez Demsar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* 7 (2006), 1–30. http://www.jmlr.org/papers/v7/demsar06a.html

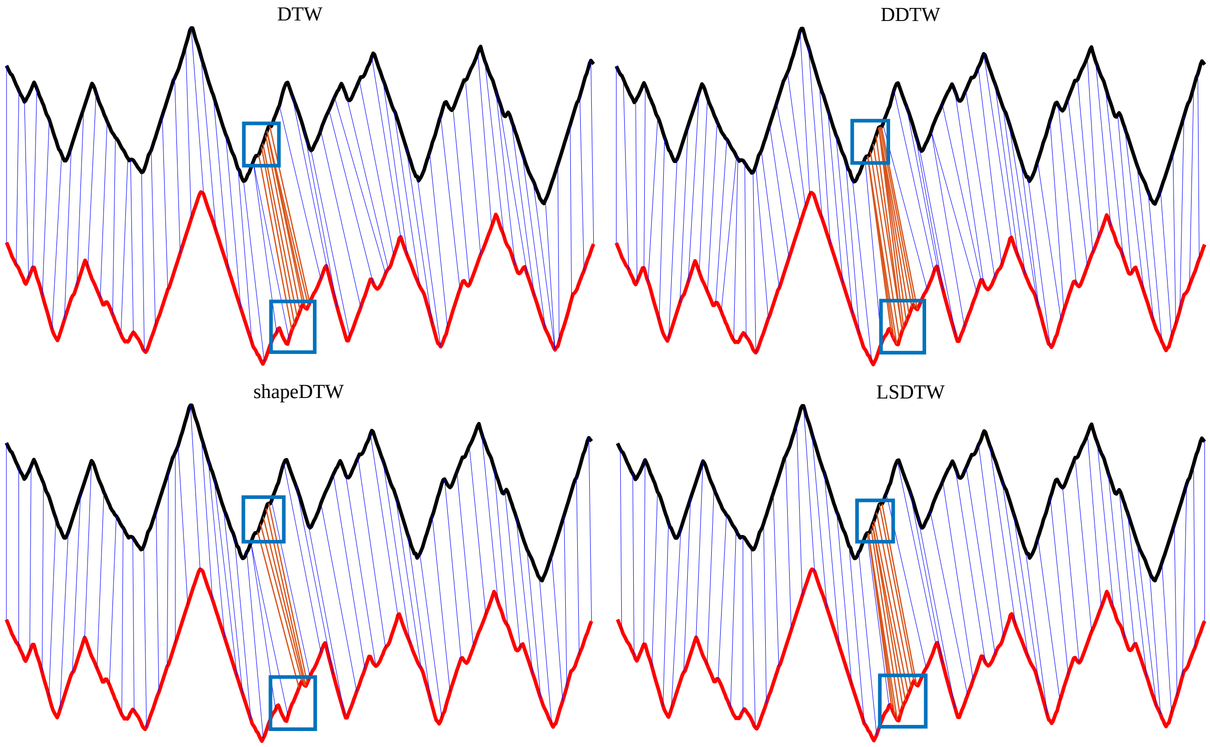

Figure 15: Sequence alignments of DTW, DDTW, shapeDTW and LSDTW on *OSULeaf*. The boxed areas have similar shape features (i.e. two peaks and valleys) with large difference in amplitude. Our LSDTW captures these features locally and globally.

Houtao Deng, George C. Runger, Eugene Tuv, and Vladimir Martyanov. 2013. A time series forest for classification and feature extraction. *Inf. Sci.* 239 (2013), 142–153. https://doi.org/10.1016/j.ins.2013.02.030

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre Alain Muller. 2018. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery* (2018), 1–47.

Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. 2014. Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 392–401.

Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. 2014. Classification of time series by shapelet transformation. *Data Min. Knowl. Discov.* 28, 4 (2014), 851–881.

Fumitada Itakura. 1975. Minimum prediction residual principle applied to speech recognition. *IEEE Trans. Acoust., Speech, Signal Processing* 23, 1 (1975), 67–72.

Youngseon Jeong, Myong Kee Jeong, and Olufemi A. Omitaomu. 2011. Weighted dynamic time warping for time series classification. *Pattern Recognition* 44, 9 (2011), 2231–2240. https://doi.org/10.1016/j.patcog.2010.09.022

Rohit J Kate. 2016. Using dynamic time warping distances as features for improved time series classification. *Data Min. Knowl. Discov.* 30, 2 (2016), 283–312.

Eamonn J. Keogh and Michael J. Pazzani. 2001. Derivative Dynamic Time Warping. In *Proceedings of the First SIAM International Conference on Data Mining, SDM 2001.* 1–11. https://doi.org/10.1137/1.9781611972719.1

Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*.

Xiaosheng Li and Jessica Lin. 2017. Linear Time Complexity Time Series Classification with Bag-of-Pattern-Features. In *Data Mining (ICDM), 2017 IEEE International Conference on*. IEEE, 277–286.

Jessica Lin, Rohan Khade, and Yuan Li. 2012. Rotation-invariant similarity in time series using bag-of-patterns representation. *J. Intell. Inf. Syst.* 39, 2 (2012), 287–315. https://doi.org/10.1007/s10844-012-0196-5

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

Pierre-Francois Marteau. 2009. Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 2 (2009), 306–318. https://doi.org/10.1109/TPAMI.2008.76

Thach Le Nguyen, Severin Gsponer, and Georgiana Ifrim. 2017. Time series classification by sequence learning in all-subsequence space. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 947–958.

Thach Le Nguyen, Severin Gsponer, Iulia Ilie, and Georgiana Ifrim. 2018. Interpretable Time Series Classification using All-Subsequence Learning and Symbolic Representations in Time and Frequency Domains. *arXiv preprint arXiv:1808.04022* (2018).

Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. 2002. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 7 (2002), 971–987. https://doi.org/10.1109/TPAMI.2002.1017623

Yu Qiao and Makoto Yasuhara. 2006. Affine Invariant Dynamic Time Warping and its Application to Online Rotated Handwriting Recognition. In *18th International Conference on Pattern Recognition (ICPR 2006)*. 905–908. https://doi.org/10.1109/ICPR.2006.228

Thanawin Rakthanmanon and Eamonn Keogh. 2013. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 668–676.

Hiroaki Sakoe and Seibi Chiba. 2003. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Trans. Acoust., Speech, Signal Processing* 26, 1 (2003), 43–49.

Patrick Schäfer. 2015. The BOSS is concerned with time series classification in the presence of noise. *Data Min. Knowl. Discov.* 29, 6 (2015), 1505–1530. https://doi.org/10.1007/s10618-014-0377-7

Patrick Schäfer and Ulf Leser. 2017. Fast and accurate time series classification with weasel. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 637–646.

Pavel Senin and Sergey Malinchik. 2013. SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model. In *2013 IEEE 13th International Conference on Data Mining, 2013*. 1175–1180. https://doi.org/10.1109/ICDM.2013.52

Joan SerrÃă, Santiago Pascual, and Alexandros Karatzoglou. 2018. Towards a universal neural network encoder for time series. (2018).

Mohan Shi, Zhihai Wang, Jidong Yuan, and Haiyang Liu. 2018. Random Pairwise Shapelets Forest. In *Pacific-asia Conference on Knowledge Discovery & Data Mining*.

Alexandra Stefan, Vassilis Athitsos, and Gautam Das. 2013. The Move-Split-Merge Metric for Time Series. *IEEE Trans. Knowl. Data Eng.* 25, 6 (2013), 1425–1438. https://doi.org/10.1109/TKDE.2012.88

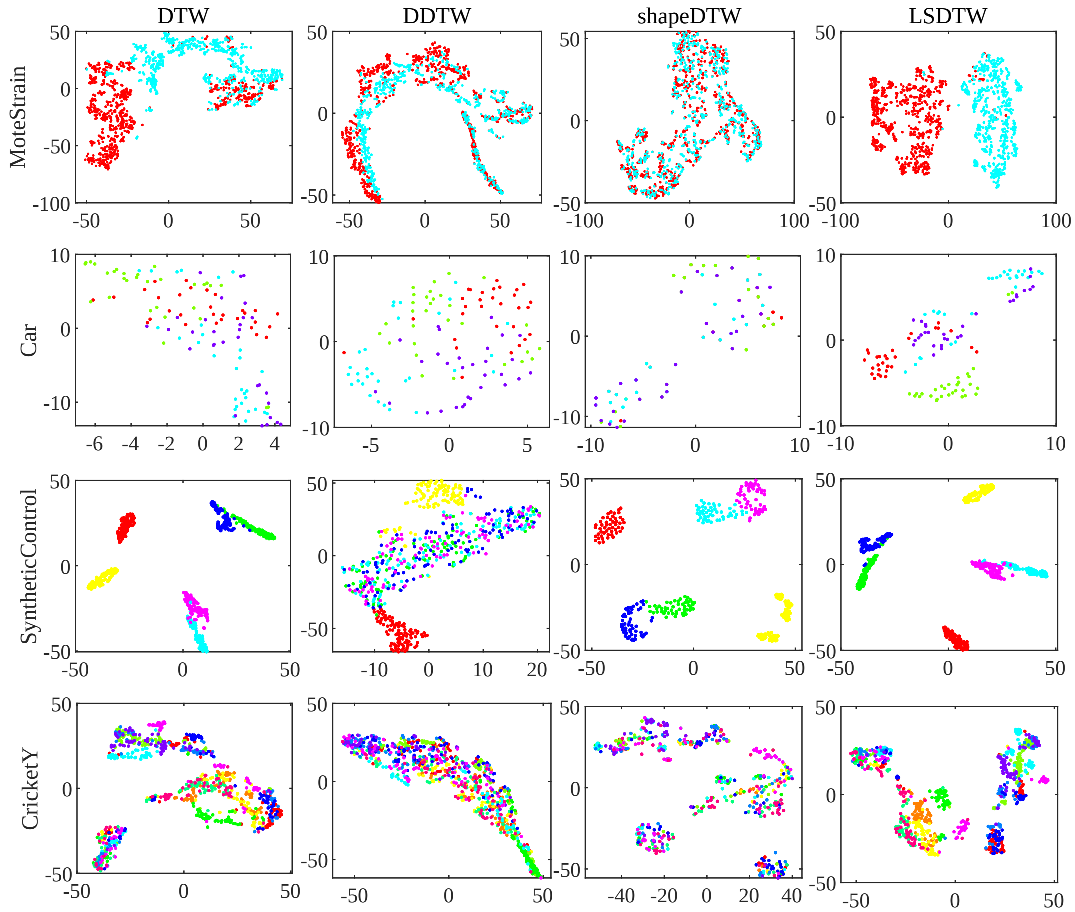

Figure 16: Underlying structures of DTW, DDTW, shapeDTW and LSDTW (from left to right) on datasets *MoteStrain*, *Car*, *SyntheticControl* and *CricketY* (from top to bottom). Different colors represent different classes in each subgraph, and each colored dot indicates one instance.

Pattreeya Tanisaro and Gunther Heidemann. 2017. Time Series Classification Using Time Warping Invariant Echo State Networks. In *IEEE International Conference on Machine Learning Applications*.

Zhiguang Wang, Weizhong Yan, and Tim Oates. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *International Joint Conference on Neural Networks*.

Jidong Yuan, Ahlame Douzal-Chouakria, Saeed Varasteh Yazdi, and Zhihai Wang. 2018. A large margin time series nearest neighbour classification under locally weighted time warps. *Knowl. Inf. Syst.* 3 (2018), 1–19.

Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. 2017. Convolutional neural networks for time series classification. *Journal of Systems Engineering & Electronics* 28, 1 (2017), 162–169.

Jiaping Zhao and Laurent Itti. 2016. Classifying Time Series Using Local Descriptors with Hybrid Sampling. *IEEE Trans. Knowl. Data Eng.* 28, 3 (2016), 623–637. https://doi.org/10.1109/TKDE.2015.2492558

Jiaping Zhao and Laurent Itti. 2018. shapeDTW: Shape Dynamic Time Warping. *Pattern Recognition* 74 (2018), 171–184. https://doi.org/10.1016/j.patcog.2017.09.020

Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J. Leon Zhao. 2014. Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. (2014).