Journal of Computational Science 76 (2024) 102235

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

An exhaustive comparison of distance measures in the classification of time series with 1NN method

Tomasz Górecki *, Maciej Łuczak, Paweł Piasecki

Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Uniwersytetu Poznańskiego 4, 61-614, Poznań, Poland

A R T I C L E I N F O

Keywords:
Time series
Classification
Distance measures
UCR archive

A B S T R A C T

Time series classification is an important and challenging problem in data analysis. With the increase in time series data availability, hundreds of algorithms have been proposed. A huge effort over the past two decades caused a significant improvement in both the efficiency and effectiveness of time series classification. There is a belief in the community that the best method is a surprisingly simple one. Even though there exist many algorithms outperforming the nearest neighbor (NN) classifier, the popularity of the latter remains stable — due to its simplicity and high performance in many domains, especially with dynamic time warping (DTW) as the distance measure. In the paper, we present an exhaustive study in which we compare the performance of different similarity measures relying on the 1NN classifier. We used the most highly cited time series distance measures used in classification (in total we compared 56 distance measures). We evaluate methods on all datasets from the UCR Time Series Classification Archive. Additionally, we perform extensive statistical comparison of the examined methods. We show that none of the distance measures is the best for all datasets, however, there is a group performing statistically significantly better than the others.

1. Introduction

Fueled by the massive growth of data and the rapidly increasing instrumentation requirements of next-generation software (DevOps monitoring data, mobile/web application event streams, industrial machine data, scientific measurements), time series methods have become more popular. Such analyses are useful in various fields, just a few:

- Financial data – sales forecasting [1], stock market analysis [2], price estimation [3].
- Weather data – temperature estimation [4], climate change [5], weather forecasting [6].
- Network data – network usage prediction [7], anomaly or intrusion detection [8], predictive maintenance [9].
- Healthcare data – census prediction [10], epileptic seizure prediction [11], patient monitoring [12].

One of the most popular and challenging methods for time series is classification. Different approaches have been proposed in the literature for time series classification, including distance-based classification methods. The k-nearest neighbor (kNN) is one of the oldest, simplest, and most accurate algorithms for classification and regression. The advantages of this classifier are essentially its simplicity, and effectiveness and it also being a parameter-free technique. It was proposed by Fix and Hodges [13] and then modified by Cover and Hart [14]. kNN method has been identified as one of the top ten methods in data mining [15]. The kNN algorithm classifies an unlabeled test sample based on the majority of similar samples among the k-nearest neighbors that are the closest to the test sample. Hence, the 1NN method classifies a new example to the class of the closest example from the training sample. The distances between the test sample and each of the training data samples are determined by a specific distance measure. The choice of the similarity measure plays a major role in the accuracy of the 1NN classifier. Hence, it represents a good basis for comparing the performance of similarity measures.

The last years (up to the year 2019) have seen increasing conviction that the 1NN method with DTW as the distance measure (1NN-DTW) is the algorithm of the first choice for most time series classification problems [16]. Some studies have strongly confirmed this idea:

- Serrà et al. [17] compare 7 similarity measures on 45 datasets from UCR archive [18]. The authors suggest that, in the set of investigated distances, there is a group of measures with no statistically significant differences: DTW, EDR, and MJC. Euclidean distance is said to perform statistically worse than TWED, DTW, EDR, and MJC, and even its performance on large datasets was not "impressive".

* Corresponding author.
E-mail addresses: tomasz.gorecki@amu.edu.pl (T. Górecki), mluczak@amu.edu.pl (M. Łuczak), pawel.piasecki@amu.edu.pl (P. Piasecki).

https://doi.org/10.1016/j.jocs.2024.102235
Received 19 August 2023; Received in revised form 27 December 2023; Accepted 8 February 2024
Available online 10 February 2024
1877-7503/© 2024 Elsevier B.V. All rights reserved.

Table 1
Comparison between recent studies for time series classification with 1NN.

Reference	#Distances	#Datasets	Percentage of UCR (as of 2023)
Serrà et al. [17]	7	45	35.16
Wang et al. [19]	13	38	29.69
Bagnall and Lines [20]	18	77	60.16
Bagnall et al. [21]	19	85	66.41
Górecki and Piasecki [22]	26	34	26.56
Górecki and Piasecki [23]	30	47	36.72
Current article	56	128	100.00

- Wang et al. [19] in an empirical study compared 1NN-DTW to the most highly cited distance measures in the literature on 38 datasets from UCR. They found that no distance measure consistently beats DTW, but DTW almost always outperforms most methods that were originally touted as superior, based on less complete empirical evaluations.
- In Bagnall and Lines [20] the authors examine the assumption that the 1NN classifier is the best technique and consider other classifiers, including kernel methods, decision trees, random forests, and others. They performed experiments on 77 datasets from UCR. Once again, the evidence strongly suggests that for time series classification 1NN-DTW is the most suitable method.
- Bagnall et al. [21] examined 19 time series classification algorithms on 85 datasets from UCR. Their results indicate that only nine of these algorithms are significantly more accurate than the 1NN-DTW classifier.
- Górecki and Piasecki [22,23] compared 26 (30) distance measures on 34 (47) datasets from UCR. They observed, that there is no measure distinctly better than the others or appropriate for a majority of datasets. On the other hand, the best average ranks were achieved by modifications of DTW distance and edit-based distances. The current work significantly develops their previous analyses. We have increased the number of datasets to 128, and the number of analyzed distances to 56. Moreover, numerous additional analyses and visualizations have been introduced.

The past three years have seen the emergence of several new methods that seem to surpass those based solely on distances, e.g., ROCKET [24], MiniROCKET [24], MultiROCKET [25], Hydra-MultiROCKET [26], HIVE-COTE version 1 (HC1) [27] and version 2 (HC2) [28] or TS-CHIEF [29]. Middlehurst et al. [30] state that Hydra-MultiROCKET and HC2 generally perform the best. So, do such methods still have a reason to exist? It seems that they do. It is important to emphasize that they are usually much cheaper computationally. Moreover, even if they achieve slightly weaker results, they are still considered simple and good baselines. Because of these, most recent research has assumed the utility of 1NN-DTW.

However, the UCR archive has recently been expanded to 128 datasets (authors also corrected the normalization of some datasets). Aspects of previous evaluations have made comparisons between algorithms difficult. The relaunch of the archive provides a timely opportunity to thoroughly evaluate algorithms on a larger number of datasets. As can be seen from the above literature review of most related works, all of the previous works have investigated either a small number of distances and similarity measures, a small number of datasets, or both (Table 1).

The problem of similarity measures is a major area of interest within the field of time series classification. There are a plethora of good time series similarity measures in the literature. To the best of our knowledge, more than 50 distance measures for time series classification have already been proposed in the literature. Such a variety may be confusing and makes it hard to find the best measure. Despite giving interesting results, the above studies take into account only some distance measures, while in the meantime, there are many more available.

Our contribution is to give an extensive comparison, supported by comprehensive statistical analysis. To the best of our knowledge, our study is the biggest available study taking into account both the number of datasets (128) and the number of distances (56). Our goal was to create a benchmark study, that could be used not only by applications of time series classification methods but as well to develop new distance measures, to assess their effectiveness.

The rest of this paper is structured as follows. We first give short descriptions of all distances that we compare in this article (Section 2). In Section 3 we describe the experimental setup, i.e. data, tools, and methodology that we used to evaluate distance measures. Section 4 contains the results of our experiment enriched with statistical analysis. At the beginning of this section, we give a short description of the statistical tools used in the comparison. In Section 5 we summarize the findings and discuss possible future work. At the end, tables with detailed results can be found.

2. Distances' classification and description

A metric or distance function is a function that defines the distance between each pair of elements of a set. Formally:

Definition 1 (*Metric*)**.** A metric on a set X is a function

$d : X \times X \to [0, \infty),$

and for all $x, y, z \in X$, the following conditions are satisfied:

1.$d(x, y) = 0 \iff x = y$ (Identity),

2.$d(x, y) = d(y, x)$ (Symmetry),

3.$d(x, y) \le d(x, z) + d(z, y)$ (Triangle inequality).

These conditions also imply the non-negativity:

$d(x, y) \ge 0$

for all $x, y \in X$. In classification with the kNN method, we can use a weaker function called semimetric (dissimilarity measure) that does not necessarily satisfy the triangle inequality.

Definition 2 (*Semimetric*)**.** A semimetric on a set X is a function

$d : X \times X \to [0, \infty),$

and for all $x, y \in X$, the following conditions are satisfied:

1.$d(x, y) \ge 0,$

2.$d(x, y) = 0 \iff x = y,$

3.$d(x, y) = d(y, x).$

Definition 3 (*Time series*)**.** By a time series $\boldsymbol{X}_N$, we will understand an ordered sequence $x_1, x_2, \dots, x_N$ of N real values. It can be also multivariate, where at each timestamp several values are obtained simultaneously.

In this work, we are only interested in the univariate time series.

Most time series classification algorithms can be divided into two categories from the perspective of similarity, i.e., whole-series-based and feature-based algorithms. Whole-series-based algorithms consider distance measures on the whole time series and use usually nearest neighbor classifier to make the classification. Feature-based algorithms transform original time series into different representations or feature vectors. Feature-based algorithms are divided into three categories, i.e., intervals-based, shapelets-based, and dictionary-based algorithms [21].

There are two main approaches for time series classification: shape-based classification and feature-based classification [31]. Shape-based classification determines the best class according to a distance measure between examples. Feature-based classification finds the best class

Table 2
L_p distances.

Distance	p	Formula
Manhattan – L_1 (MAN)	$p = 1$	$\sum_{i=1}^{N} \lvert x_i - y_i \rvert$
Euclidean – L_2 (ED)	$p = 2$	$\sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$
Minkowski – L_p (MIN)	$1 \le p < \infty$	$\sqrt[p]{\sum_{i=1}^{N} \lvert x_i - y_i \rvert^p}$
Chebyshev norm – L_∞ (INF)	$p = \infty$	$\max_{i=1,\dots,N} \lvert x_i - y_i \rvert$

according to the set of features defined for the time series (e.g. standard deviation, mean, complexity, etc.). Wang et al. [19] in their study distinguished four groups of distance measures: lock-step measures, elastic measures, threshold-based measures, and pattern-based measures. Montero and Vilar [32] proposed to group measures into four categories: model-free measures, model-based measures, complexity-based measures, and prediction-based measures. In our opinion, the most universal categorization covering almost all distances is proposed by Esling and Agón [33]: shape-based measures, edit-based measures, feature-based measures, and structure-based measures. We intend to follow the latest classification. However, we propose to distinguish one extra class - combined distances, which arose in the last few years and may be defined as distances made of a convex combination of some other distances. In this section, we list all distance measures used in this paper.

2.1. Shape-based distance measures

L_p Distances. This group of distance measures compares the overall shape of series performing only arithmetic operations on the raw values. L_p distances are directly derived from L_p norms. They are widely used mainly thanks to their simplicity and ease of computation [34]. However, their drawbacks are poor performance [35], measuring only time series of equal length, and being highly influenced by outliers, noise, scaling, or warping. Given two time series $\boldsymbol{X}_N = (x_1, x_2, \dots, x_N)$ and $\boldsymbol{Y}_N = (y_1, y_2, \dots, y_N)$ we compute them with formulas from Table 2. In our experiments for Minkowski distance, we made an arbitrary choice of $p = 3$, as results obtained for other values were roughly similar (we tested p from 3 to 20).

Additionally, let us note that such types of distances can be utilized in classic machine learning models eg. random forest, logistic regression, support vector regression, etc. Dhariyal et al. [36] showed that such methods could achieve good results, which are even comparable to state-of-the-art algorithms for time series classification.

Derivative Euclidean Distance (DED). DED [37] is a distance measure defined as

$$\mathrm{DED}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \mathrm{ED}(\boldsymbol{X}'_N, \boldsymbol{Y}'_N),$$

where

$$\boldsymbol{X}'_N = (x_{n+1} - x_n : n = 1, \dots, N-1)$$

is a discrete derivative (first) of the series $\boldsymbol{X}_N$.

Short Time Series (STS) distance. Möller-Levet et al. [38] proposed the STS distance. The objective was to define a distance that can capture differences in the shapes, defined by the relative change of expression and the corresponding temporal information, regardless of the difference in absolute values. Authors consider the time series as piecewise linear functions and measure the difference of slopes between them. There are two main advantages of STS distance. Firstly, it handles irregularly sampled series. Secondly, it can measure the similarity of shapes formed by the relative change of amplitude and the corresponding temporal information [38]. It is given by

$$\mathrm{STS}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \sqrt{\sum_{i=1}^{N-1} \left(\frac{y_{i+1} - y_i}{t_{i+1} - t_i} - \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \right)^2}.$$

Dynamic Time Warping (DTW) distance. DTW [39] is one of the most popular distance measures due to its ability to deal with the warping of the time axis. DTW aims to find the optimal alignment between two series by looking for the shortest warping path in a distance matrix. To compute the DTW distance, we first construct a cost (distance) matrix $\boldsymbol{D}$, where the (i, j)-th element is given by:

$$d(x_i, y_j) = (x_i - y_j)^2.$$

Then, DTW distance is found as a path through the matrix with minimal overall cost (warping cost), which is usually written as:

$$\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \min \sqrt{\sum_{k=1}^{K} w_k},$$

where w_k is the element of matrix $\boldsymbol{D}$ that is also the kth element of a warping path W (Fig. 1). To detect the optimal path, we use a dynamic programming algorithm, with time complexity $O(NM)$. To speed up computations even more, several lower bounding and temporal constraints techniques have been proposed. In Section 4, we denote the DTW constrained by the Sakoe-Chiba band as "DTWc" and we use the window size as in Dau et al. [18]. For more details about DTW, we refer to Bagnall et al. [21], Keogh and Ratanamahatana [40] and Mori et al. [41]. Euclidean distance is a special case of DTW — the only considered warping path is built out of diagonal elements of the matrix $\boldsymbol{D}$ and it is defined only for time series of equal length (Fig. 1). However, in general, DTW does not satisfy triangle inequality and hence is not a metric.

Derivative Dynamic Time Warping (DDTW) distance. DDTW [37] is a distance measure defined

$$\mathrm{DDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \mathrm{DTW}(\boldsymbol{X}'_N, \boldsymbol{Y}'_M),$$

where

$$\boldsymbol{X}'_N = (x_{n+1} - x_n : n = 1, \dots, N-1)$$

is a discrete derivative (first) of the series $\boldsymbol{X}_N$.

Second Derivative Dynamic Time Warping (2DDTW) distance. 2DDTW [42] is a distance measure defined

$$\mathrm{2DDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \mathrm{DTW}(\boldsymbol{X}''_N, \boldsymbol{Y}''_M),$$

where

$$\boldsymbol{X}''_N = (\boldsymbol{X}'_N)'$$

is a discrete second derivative of the series $\boldsymbol{X}_N$.

Integral Dynamic Time Warping (IDTW) distance. IDTW [43] is a distance measure defined

$$\mathrm{IDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \mathrm{DTW}(\mathrm{I}(\boldsymbol{X}_N), \mathrm{I}(\boldsymbol{Y}_M)),$$

where

$$\mathrm{I}(\boldsymbol{X}_N) = \left\{ \sum_{i=1}^{n} x_i : n = 1, \dots, N, \right\}$$

is a discrete integral of the series $\boldsymbol{X}_N$.

Shape Dynamic Time Warping (shapeDTW) distance. shapeDTW [44] consists of two major steps: encode local structures by shape descriptors and align descriptor sequences by DTW. Concretely, we sample a subsequence from each temporal point and further encode it by some shape descriptor. As a result, the original time series is converted into a descriptor sequence of the same length. Then we align two descriptor sequences by DTW and transfer the found warping path to the original time series. Two main descriptors examined in [44] were Raw-Subsequence and HOG1D. This gives two shapeDTW distance variants:

$$\mathrm{shapeDTW}_{\mathrm{RawSub}} \quad \text{and} \quad \mathrm{shapeDTW}_{\mathrm{HOG1D}}.$$

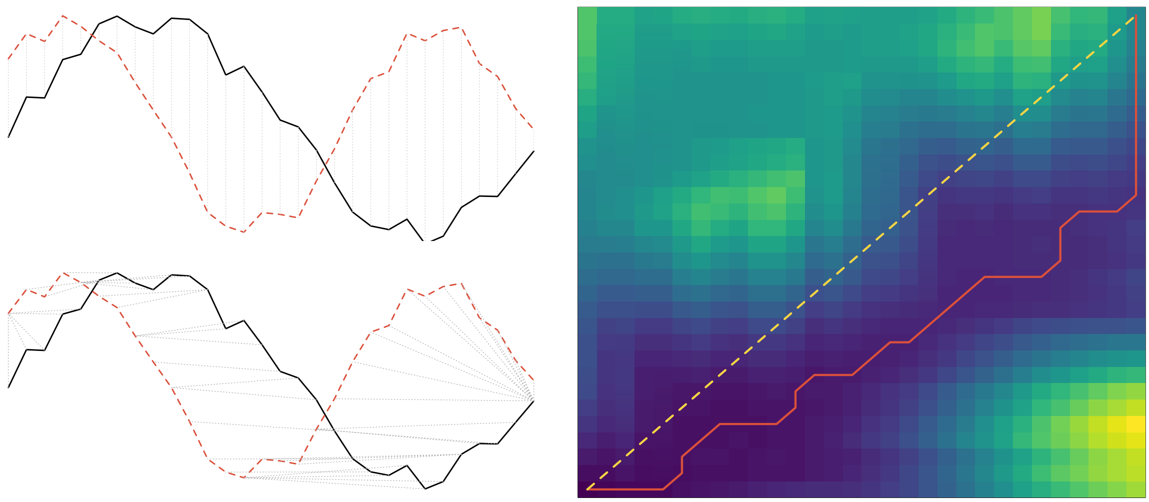

Fig. 1. Top left: Euclidean distance between two time series that are similar but shifted is large. Bottom left: Introducing flexibility by DTW's non-linear alignment allows for lowering the distance and as a result, we consider time series more similar. Right: The heat map visualizes the cost matrix $\boldsymbol{D}$. The red solid line is the optimal warping path, while the yellow dashed line corresponds to Euclidean distance.

Complexity-invariant dissimilarity measure. Many dissimilarity measures tend to put time series with high complexity levels further apart than simple ones [45]. To fix this distortion, a correction factor has been proposed by Batista et al. [45]. A general complexity-invariant dissimilarity measure (CID) is defined as follows

$$\mathrm{CID}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \mathrm{CF}(\boldsymbol{X}_N, \boldsymbol{Y}_N) \cdot d(\boldsymbol{X}_N, \boldsymbol{Y}_N), \tag{1}$$

where $d(\boldsymbol{X}_N, \boldsymbol{Y}_N)$ is a distance which may be chosen (in our experiment we use Euclidean distance) and $\mathrm{CF}(\boldsymbol{X}_N, \boldsymbol{Y}_N)$ is a complexity correction factor defined as

$$\mathrm{CF}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \frac{\max\{\mathrm{CE}(\boldsymbol{X}_N), \mathrm{CE}(\boldsymbol{Y}_N)\}}{\min\{\mathrm{CE}(\boldsymbol{X}_N), \mathrm{CE}(\boldsymbol{Y}_N)\}}, \tag{2}$$

where $\mathrm{CE}(\boldsymbol{X}_N)$ stands for a complexity estimator of $\boldsymbol{X}_N$. From Eq. (1), we can observe, that when the complexity of both time series is equal, we get

$$\mathrm{CID}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = d(\boldsymbol{X}_N, \boldsymbol{Y}_N)$$

and from Eq. (2) that an increase of complexity difference results in increase of distance between time series. As a complexity estimator Batista et al. [45] proposed

$$\mathrm{CE}(\boldsymbol{X}_N) = \sqrt{\sum_{i=1}^{N-1} (x_i - x_{i+1})^2}.$$

The time complexity of the entire distance is $O(N)$ (for Euclidean distance as d).

Matrix Profile distance (MPdist). MPdist has been introduced by Gharghabi et al. [46] as a robust alternative for widely used Euclidean distance and Dynamic Time Warping, especially suitable for difficult domains. The main advantage of MPdist is the way it compares time series — it seeks how much they have in common in terms of subsequences, instead of being based on raw values. It is highly desirable in the case of datasets where pattern matters more than e.g. amplitude or offset. The idea of MPdist relies on a versatile algorithm: Matrix Profile [47], which cleverly deals with the *all-pairs-similarity-search* problem. The detailed procedure of computing MPdist is given by Gharghabi et al. [46], however, we can briefly describe it in 4 steps:

1. Given two time series $\boldsymbol{X}_N$ and $\boldsymbol{Y}_M$ compute all-subsequences sets:
$$\boldsymbol{A}_X = \{(x_1, \dots, x_L), (x_2, \dots, x_{L+1}), \dots, (x_{N-L+1}, \dots, x_N)\}$$
$$\boldsymbol{A}_Y = \{(y_1, \dots, y_L), (y_2, \dots, y_{L+1}), \dots, (y_{M-L+1}, \dots, y_M)\}$$
for $\boldsymbol{X}_N, \boldsymbol{Y}_M$ respectively, for subsequences of length L.
2. For each element of $\boldsymbol{A}_X$ compute the distance to the nearest neighbor in $\boldsymbol{A}_Y$ and in a result create a new time series $\boldsymbol{P}_{\mathrm{XY}}$ of length $N - L + 1$. Compute analogously $\boldsymbol{P}_{\mathrm{YX}}$.
3. Concatenate $\boldsymbol{P}_{\mathrm{XY}}$ and $\boldsymbol{P}_{\mathrm{YX}}$ to get $\boldsymbol{P}_{\mathrm{XYYX}}$.
4. Compute MPdist as below:
$$\mathrm{MPdist}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \begin{cases} \mathrm{k}^{th}\text{value of sorted } \boldsymbol{P}_{\mathrm{XYYX}}, & \text{if } |\boldsymbol{P}_{\mathrm{XYYX}}| > k \\ \max(\boldsymbol{P}_{\mathrm{XYYX}}), & \text{if } |\boldsymbol{P}_{\mathrm{XYYX}}| \le k, \end{cases}$$
where k is an arbitrarily chosen parameter with suggestion made by [46] to set k to be equal 5% of $2 \times N$.

Note that this implies that when the length of a subsequence is equal to the length of full time series ($L = N$), the MPdist degenerates to the Euclidean distance. The time complexity of MPdist for two equal-length time series is $O(N^2)$ in the worst case (when L is much shorter than N). For the moment, there is no clear rule on how to choose the sole parameter L, thus in the computational part, it was learned from the training datasets.

Time Alignment Measurement (TAM) distance. TAM [48] has been proposed as an alternative for DTW. The latter one is used mostly for time series warped in time, however, it measures only amplitude similarity. TAM distance aims to reflect the degree of time warping between time series in terms of phase shifts observed on the cost matrix with the optimal warping path, obtained in the same way as in DTW. Considering series $\boldsymbol{X}_N$ which is plotted on the x-axis, we can define the total time $\overleftarrow{\theta}_{xy}$ which $\boldsymbol{Y}_M$ is delayed to $\boldsymbol{X}_N$, as many vertical segments in the warping path (such where $w_{k+1} - w_k = (0, 1)$). The total time $\vec{\theta}_{xy}$ which series $\boldsymbol{Y}_M$ is in advance to $\boldsymbol{X}_N$ is measured as many horizontal segments in the warping path (such where $w_{k+1} - w_k = (1, 0)$). When $\boldsymbol{Y}_M$ is delayed or in advance compared to $\boldsymbol{X}_N$, we say that series is *out of phase*. Subsequences are *in phase* when there is no time warping (diagonal segments in the warping path where $w_{k+1} - w_k = (1, 1)$). The number of diagonal segments we denote as $\overline{\theta}_{xy}$. Given that, we define the fractions of advance ($\overrightarrow{\psi}$), delay ($\overleftarrow{\psi}$), and phase ($\overline{\psi}$) segments to length of series as:

$$\overrightarrow{\psi} = \frac{\vec{\theta}_{xy}}{N}, \qquad \overleftarrow{\psi} = \frac{\overleftarrow{\theta}_{xy}}{M}, \qquad \overline{\psi} = \frac{\overline{\theta}_{xy}}{\min\{N, M\}}.$$

Eventually, the TAM distance is formulated below:

$$\mathrm{TAM}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \overrightarrow{\psi} + \overleftarrow{\psi} + (1 - \overline{\psi}).$$

Unfortunately, TAM distance does not satisfy the condition of identity of indiscernibles, so it cannot be considered as a metric. Finally, let us observe that TAM distance takes values from 0 (series equal in the temporal domain) to 3 (out of phase all along the warping path).

2.2. Edit-based distance measures

Edit-based distances calculate the minimum number of operations – such as delete, insert, replace, split, and merge (as referred to below) – required to transform one series into another. They were initially proposed to measure the similarity between two sequences of strings and use the minimal number of the so-called edit operations necessary to transform one series into another. It is worth noting, that due to using edit operations, all of the distances presented in the subsection allow comparing time series of different lengths.

Longest Common Subsequence (LCSS) distance. Vlachos et al. [49] proposed the LCSS distance and measures the similarity between time series in terms of the longest common subsequence, with the addition that gaps and unmatched regions are permitted. LCSS is partly robust to noise and we expect that it should be more accurate than DTW in the presence of outliers and noise. The LCSS has two constant parameters: δ and ε. The $\delta > 0$ parameter controls the size of the window for matching points between two series. It may be considered as a warping window and is usually set to a percentage of the series length. The parameter $\varepsilon > 0$ is the matching threshold: two points from two series are considered to match if their distance is less than ε. Since the scope of interest of this paper is not to explore how to find the optimal values of the parameters, these values were selected under the proposal of Górecki [50]. We set the δ parameter to 100% and as the ε parameter, we used a value equal to the smallest standard deviation between the two sequences that were examined at any time. Let us define

$$L(i,j) = \begin{cases} 0 & \text{for } i = 0 \\ 0 & \text{for } j = 0 \\ 1 + L[i-1, j-1] & \text{for } |x_i - y_j| < \varepsilon \\ & \text{and } |i-j| \le \delta \\ \max(L[i-1,j], L[i,j-1]) & \text{in other cases.} \end{cases} \tag{3}$$

Suppose we have two time series: $\boldsymbol{X}_N$ and $\boldsymbol{Y}_M$, using Eq. (3), the LCSS distance may be computed as [51]:

$$\mathrm{LCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \frac{N + M - 2L(N,M)}{N+M}.$$

This measure takes values from $(1 - 2\frac{\min(N,M)}{N+M})$ to 1. For two series of equal length, it takes values from 0 to 1. LCSS does not satisfy the triangle inequality [49], so it is not a distance metric.

Derivative LCSS (DLCSS) distance. DLCSS proposed in [52] is a distance measure defined as

$$\mathrm{DLCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \mathrm{LCSS}(\boldsymbol{X}'_N, \boldsymbol{Y}'_M),$$

where

$$\boldsymbol{X}'_N = (x_{n+1} - x_n : n = 1, \dots, N-1)$$

is a discrete derivative (first) of the series $\boldsymbol{X}_N$.

Second Derivative LCSS (2DLCSS) distance. 2DLCSS proposed in [52] is a distance measure defined as

$$\mathrm{2DLCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \mathrm{LCSS}(\boldsymbol{X}''_N, \boldsymbol{Y}''_M),$$

where

$$\boldsymbol{X}''_N = (\boldsymbol{X}'_N)'$$

is a discrete second derivative of the series $\boldsymbol{X}_N$.

Edit Distance on Real Sequence (EDR). The EDR is an adaptation of the edit distance that finds the minimal number of edit operations (delete, insert, replace) to convert one series into another [53]. Similarly to LCSS, EDR permits gaps and unmatched regions but penalizes such occurrences with a value equal to their length. EDR can be computed using dynamic programming as follows:

$$\mathrm{EDR}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \begin{cases} N & \text{for } i = 0 \\ M & \text{for } j = 0 \\ \min\{\mathrm{EDR}(\mathrm{Rest}(\boldsymbol{X}_N), & \\ \mathrm{Rest}(\boldsymbol{Y}_M)) + d_{\mathrm{edr}}(x_1, y_1), & \\ \mathrm{EDR}(\mathrm{Rest}(\boldsymbol{X}_N), \boldsymbol{Y}_M) + 1, & \\ \mathrm{EDR}(\boldsymbol{X}_N, \mathrm{Rest}(\boldsymbol{Y}_M)) + 1\} & \text{otherwise,} \end{cases}$$

where

$$\mathrm{Rest}(\boldsymbol{X}_N) = (x_2, x_3, \dots, x_N)$$

and d_{edr} stands for the distance between two points in the series computed along to the rule: if x_i and y_i are closer to each other in the absolute sense than ε, it is equal to 1. Otherwise, it is equal to 0. In the calculations of EDR distance, we used only one set of parameters, to reduce the optimization step: $\varepsilon = 0.1, \sigma = 25\%$ (a Sakoe-Chiba windowing constraint) of series' length.

Edit Distance with Real Penalty (ERP). The third variation of edit distance is the ERP [54] that may be considered as a combination of DTW and EDR. It uses the L_1 distance between elements of the time series as the penalty for local shifting of the time series. Penalization is done by setting a constant g and adding the Euclidean distance of the unmatched points to g. The ERP measure is given by

$$\mathrm{ERP}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \begin{cases} \sum_{i=1}^{N} |y_i - g| & \text{if } M - 1 = 0 \\ \sum_{i=1}^{M} |x_i - g| & \text{if } N - 1 = 0 \\ \min\{\mathrm{ERP}(\mathrm{Rest}(\boldsymbol{X}_N), & \\ \mathrm{Rest}(\boldsymbol{Y}_M)) + d_{\mathrm{erp}}(x_1, y_1), & \\ \mathrm{ERP}(\mathrm{Rest}(\boldsymbol{X}_N), \boldsymbol{Y}_M) + 1, & \\ \mathrm{ERP}(\boldsymbol{X}_N, \mathrm{Rest}(\boldsymbol{Y}_M)) + 1\} & \text{otherwise,} \end{cases}$$

where $\mathrm{Rest}(\boldsymbol{X}_N)$ and d_{erp} are computed the same as in case of EDR. In the experimental part of our research, for EDR distance, we used constantly only one set of parameters, to reduce the optimization step: $g = 0.1,\ \sigma = 25\%$ of series' length.

Move-Split-Merge distance (MSM). The Move-Split-Merge distance [55], similarly to LCSS, EDR, and ERP, also uses a set of operations that can transform one-time series to another, but the set is slightly different: Move (renamed substitute) — that sets the value of the new element, Split — that repeats a value twice and Merge — that merges two successive equal values into one. Each of the operations has assigned a specific cost, which is referred to below. Given time series $\boldsymbol{X}_N = (x_1, \dots, x_N)$, the Move operation, and its cost, are given by:

$$\mathrm{Move}_{i,v}(\boldsymbol{X}_N) = (x_1, \dots, x_{i-1}, x_i + v, x_{i+1}, \dots, x_N),$$
$$\mathrm{Cost}(\mathrm{Move}_{i,v}) = |v|.$$

The Split operation, and its cost, are given by:

$$\mathrm{Split}_i(\boldsymbol{X}_N) = (x_1, \dots, x_{i-1}, x_i, x_i, x_{i+1}, \dots, x_N),$$
$$\mathrm{Cost}(\mathrm{Split}_i) = c.$$

The Merge operation, and its cost, are defined as:

$$\mathrm{Merge}_i(\boldsymbol{X}_N) = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N),$$
$$\mathrm{Cost}(\mathrm{Merge}_i) = c.$$

The MSM distance is defined as the lowest sum of the total cost of transformation of one time series to another and in practice can be computed using dynamic programming. The MSM is a metric. It contains one free parameter c, which was set during the experiment to one of the values suggested by authors [55]: 0.1.

2.3. Feature-based distance measures

Feature-based distances measure time series through some of their aspects, trying to extract certain features. They may be based e.g. on some association measure or transformation. It is the most numerous and the broadest group of dissimilarity measures.

Distance based on pearson's correlation. One of the basic ideas for feature-based distance is to utilize Pearson's correlation coefficient between two time series and create a measure based on that. Two such measures have been proposed so far [56]:

$$d_{\mathrm{PC1}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \sqrt{\left(\frac{1-PC}{1+PC}\right)^{\beta}},$$

$$d_{\mathrm{PC2}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \sqrt{2(1-PC)},$$

where PC denotes Pearson's correlation coefficient, β is a positive parameter (the parameter β enables the control of the rapid decrease in distance). To the best of our knowledge, recommendations on the choice of β are still not established, thus – in the experimental part – we limited ourselves to examining only the parameter-free PC2 distance ($\beta = 1$).

Distance based on the cross-correlation. Based on cross-correlation, [57] defined the following distance measure

$$d_{\mathrm{CC}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \sqrt{\frac{(1 - CC_0(\boldsymbol{X}_N, \boldsymbol{Y}_N))}{\sum_{k=1}^{N-1} CC_k(\boldsymbol{X}_N, \boldsymbol{Y}_N)}},$$

where $\mathrm{CC}_k(X, \boldsymbol{Y}_N)$ is the cross correlation between two series at lag k.

Autocorrelation-based and partial autocorrelation-based distances. Let

$$\hat{\boldsymbol{\rho}}_{X_N} = (\hat{\rho}_{1,X_N}, \ldots, \hat{\rho}_{L,X_N})^T$$

and

$$\hat{\boldsymbol{\rho}}_{Y_M} = (\hat{\rho}_{1,Y_M}, \ldots, \hat{\rho}_{L,Y_M})^T$$

be the estimated autocorrelation vectors of $\boldsymbol{X}_N, \boldsymbol{Y}_M$ (respectively), for some L such that $\hat{\rho}_{i,X_N}, \hat{\rho}_{i,Y_M} \approx 0$ for $i > L$. [58] proposed the following distance:

$$d_{\mathrm{ACF}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \sqrt{(\hat{\boldsymbol{\rho}}_{X_N} - \hat{\boldsymbol{\rho}}_{Y_M})^N \boldsymbol{\Omega} (\hat{\boldsymbol{\rho}}_{X_N} - \hat{\boldsymbol{\rho}}_{Y_M})},$$

where $\boldsymbol{\Omega}$ is a matrix of weights, which define the importance of correlation at different lags. We set $\boldsymbol{\Omega}$ as an identity matrix (in this case, d_{ACF} becomes the Euclidean distance between the estimated autocorrelation functions [32]) and $L = 50$.

A similarity measure based on partial autocorrelation function (PACF) may be defined analogously, taking PACFs instead of ACFs.

An adaptive dissimilarity index combining temporal correlation and raw value behaviors. The first-order temporal correlation coefficient is defined by

$$\mathrm{CORT}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \frac{\sum_{t=1}^{N-1}(x_{t+1} - x_N)(y_{t+1} - y_N)}{\sqrt{\sum_{t=1}^{N-1}(x_{t+1} - x_N)^2}\sqrt{\sum_{t=1}^{N-1}(y_{t+1} - y_N)^2}}.$$

The CORT coefficient reflects the dynamic behavior of the series [32] and is similar to Pearson's coefficient. It belongs to the interval $[-1, 1]$ and $\mathrm{CORT}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = 1$ indicates similar behavior of series $\boldsymbol{X}_N$ and $\boldsymbol{Y}_N$, which means that in any interval $[t_i, t_{i+1}]$ they increase or decrease simultaneously. The value of -1 implies opposite behavior, while the value of 0 shows no stochastically linear dependence. Chouakria and Nagabhushan [59] proposed a dissimilarity measure utilizing CORT as

$$d_{\mathrm{CORT}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \phi_k[\mathrm{CORT}(\boldsymbol{X}_N, \boldsymbol{Y}_N)] \cdot d(\boldsymbol{X}_N, \boldsymbol{Y}_N),$$

where ϕ_k is an adaptive tuning function to automatically modulate a conventional data distance according to the temporal correlation. [59] proposed

$$\phi_k(x) = \frac{2}{1 + \exp(kx)}, k \geq 0.$$

The advantage of d_{CORT} distance is that it measures both the proximity of observations and temporal correlation for the behavior proximity estimation [32]. This index encompasses both the traditional measure for proximity with respect to values and the temporal correlation for proximity in terms of behavior. The parameter k adjusts the contributions of proximity with respect to values and behavior in the calculation of the d_{CORT}. With an increase in k, the influence of proximity with respect to behavior on the d_{CORT} becomes more significant, while the impact of proximity with respect to values diminishes. Specifically, at $k = 2$, the proximity in terms of behavior accounts for 76.2% of d_{CORT}, while the proximity in terms of values contributes 23.8% to d_{CORT} [59]. In the computational part (Section 4), we used Euclidean distance as $d(\boldsymbol{X}_N, \boldsymbol{Y}_N)$, however it could be any other distance measure (authors also used DTW and Fréchet). We used $k = 2$.

Fourier Coefficients (FC) based distance. An interesting approach to time series comparison seems to measure differences after some transformation. A simple approach is to compare coefficients derived from the Discrete Fourier Transform. The value of these coefficients reflects the associated frequency, which allows us to refer to the frequency domain instead of the time domain.

As noted by [60], in the case of many time series, most of the information is kept in their first n coefficients, where $n < \frac{N}{2} + 1$. Based on this information, we adopted $n = \lfloor \frac{N}{2} \rfloor + 1$. Given two time series $\boldsymbol{X}_N$ and $\boldsymbol{Y}_N$ with Fourier Coefficients (respectively) $(a_0, b_0), \ldots, (a_{\frac{N}{2}}, b_{\frac{N}{2}})$, $(a'_0, b'_0), \ldots, (a'_{\frac{N}{2}}, b'_{\frac{N}{2}})$ we formally define the distance as

$$\mathrm{FC}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \sqrt{\sum_{i=0}^{n}((a_i - a'_i)^2 + (b_i - b'_i)^2)}.$$

TQuest distance. [61] proposed a distance measure based on Threshold Queries, using a given τ parameter as a threshold to transform a time series into a sequence of timestamps, when the threshold is crossed.

$$\mathrm{TQ}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \frac{1}{|S(\boldsymbol{X}_N, \tau)|} \sum_{s \in S(\boldsymbol{X}_N, \tau)} \min_{s' \in S(\boldsymbol{Y}_N, \tau)} d(s, s') + \frac{1}{|S(\boldsymbol{Y}_N, \tau)|} \sum_{s' \in S(\boldsymbol{Y}_N, \tau)} \min_{s \in S(\boldsymbol{X}_N, \tau)} d(s', s),$$

where

$$S(\boldsymbol{X}_N, \tau) = \{(t_i, t_{i+1}) : \forall_{t_i \leq t \leq t_{i+1}} X_N \geq \tau\}$$

and the distance between two intervals $s = (s_l, s_u)$, $s' = (s'_l, s'_u)$ is defined as follows

$$d(s, s') = \sqrt{(s_l - s'_l)^2 + (s_u - s'_u)^2}.$$

It is an interesting feature extraction idea, but highly dependent on the user's specialist knowledge, as the key τ parameter must be set. In the case of parameter choice, we followed remarks made by Ding et al. [62] with the simplification, that we picked up mean value. The full construction of the distance, the formula, and a synthetic description are given by Mori et al. [41].

Periodogram-based distances. Let

$$I_{X_N}(\lambda_k) = N^{-1} \left| \sum_{k=1}^{N} X_k e^{-it\lambda_k} \right|^2$$

and

$$I_{Y_N}(\lambda_k) = N^{-1} \left| \sum_{k=1}^{N} Y_k e^{-it\lambda_k} \right|^2$$

be the periodograms of $\boldsymbol{X}_N$ and $\boldsymbol{Y}_N$ (respectively) at frequencies $\lambda_k = 2\pi k/N$, where $k = 1, \ldots, n$ and $n = \lfloor N/2 \rfloor$ (where $\lfloor N/2 \rfloor$ is the largest integer less or equal to $N/2$). Based on it, Caiado et al. [63] proposed the Euclidean distance between the periodogram coordinates

$$d_{\mathrm{P}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \sqrt{\sum_{k=1}^{n}(I_{X_N}(\lambda_k) - I_{Y_N}(\lambda_k))^2}.$$

In case we are not interested in scale but rather on its correlation structure, we can use a normalized version of periodogram (given here for $\boldsymbol{X}_N$ but for $\boldsymbol{Y}_N$, of course, similar):

$$NI_{\boldsymbol{X}_N}(\lambda_k) = \frac{I_{\boldsymbol{X}_N}(\lambda_k)}{\hat{\gamma}_{0,\boldsymbol{X}_N}},$$

where $\hat{\gamma}_{0,\boldsymbol{X}_N}$ is the sample variance of $\boldsymbol{X}_N$. Thus,

$$d_{\mathrm{NP}}(\boldsymbol{X}_N,\boldsymbol{Y}_N) = \sqrt{\sum_{k=1}^{n}(NI_{\boldsymbol{X}_N}(\lambda_k) - NI_{\boldsymbol{Y}_N}(\lambda_k))^2}.$$

As suggested by Montero and Vilar [32], because the variance of the periodogram ordinates is proportional to the spectrum value at the corresponding frequencies, it may be useful to use the logarithm of the normalized periodogram:

$$d_{\mathrm{LNP}}(\boldsymbol{X}_N,\boldsymbol{Y}_N) = \sqrt{\sum_{k=1}^{n}(\log NI_{\boldsymbol{X}_N}(\lambda_k) - \log NI_{\boldsymbol{Y}_N}(\lambda_k))^2}.$$

Also, we can consider distance measures utilizing the cumulative version of the periodogram — the integrated periodogram. Casado de Lucas [64] argues that, due to some properties of the integrated periodogram, it presents several advantages over the one based on the raw periodogram. Casado de Lucas [64] proposed the following formula:

$$d_{\mathrm{IP}}(\boldsymbol{X}_N,\boldsymbol{Y}_N) = \int_{-\pi}^{\pi}|F_{\boldsymbol{X}_N}(\lambda) - F_{\boldsymbol{Y}_N}(\lambda)|d\lambda,$$

where

$$F_{\boldsymbol{X}_N}(\lambda_j) = C^{-1}_{\boldsymbol{X}_N}\sum_{i=1}^{j} I_{\boldsymbol{X}_N}(\lambda_i)$$

and

$$F_{\boldsymbol{Y}_N}(\lambda_j) = C^{-1}_{\boldsymbol{Y}_N}\sum_{i=1}^{j} I_{\boldsymbol{Y}_N}(\lambda_i),$$

with

$$C_{\boldsymbol{X}_N} = \sum_i I_{\boldsymbol{X}_N}(\lambda_i),$$

$$C_{\boldsymbol{Y}_N} = \sum_i I_{\boldsymbol{Y}_N}(\lambda_i).$$

Dissimilarity measures based on non-parametric spectral estimators. Contrary to parametric spectral methods, utilizing non-parametric spectral estimators in distance measures gives us the versatility to approximate spectra and thus we can deal with time series from different domains. Here, we consider only local linear smoothing techniques as an approximation method for log-spectra, because of their good theoretical and practical properties [65].

Kakizawa et al. [66] proposed a general spectral disparity measure between two time series as

$$d_{\mathrm{LLR}}(\boldsymbol{X}_N,\boldsymbol{Y}_N) = \frac{1}{4\pi}\int_{-\pi}^{\pi}\tilde{W}\left(\frac{f_{\boldsymbol{X}_N}(\lambda)}{f_{\boldsymbol{Y}_N}(\lambda)}\right)d\lambda,$$

where $f_{\boldsymbol{X}_N}$ and $f_{\boldsymbol{Y}_N}$ are spectral densities of $\boldsymbol{X}_N$, $\boldsymbol{Y}_N$ and

$$\tilde{W} = W(x) + W(x^{-1}),$$

$$W(x) = \log(\alpha x + (1-\alpha)) - \alpha \log x$$

with $0 < \alpha < 1$. W is a divergence function that results in d_{LLR} satisfies the quasi-distance property. However, d_{LLR} is not a real distance as it is not symmetric and does not satisfy the triangle inequality.

We do not know the "true" spectra $f_{\boldsymbol{X}_N}$ and $f_{\boldsymbol{Y}_N}$, so we have to estimate them. Depending on the estimation method, we denote $d_{\mathrm{LLR\_DLS}}$ when the spectra are obtained by local linear smoothers of the periodograms via least squares. If spectral densities are computed by the exponential transformation of local linear smoothers of the log-periodograms with maximum local likelihood criterion, we denote the distance as $d_{\mathrm{LLR\_LK}}$.

Alternatively, Díaz and Vilar [67] proposed the two following distances, differing in terms of discrepancy measure used for comparison of the two log-spectra. The first one is defined as

$$d_{\mathrm{GLK}}(\boldsymbol{X}_N,\boldsymbol{Y}_N) = \sum_{k=1}^{N}\left[Z_k - \hat{\mu}(\lambda_k) - 2\log(1+e^{Z_k-\hat{\mu}(\lambda_k)})\right] - \sum_{k=1}^{n}\left[Z_k - 2\log(1+e^{Z_k})\right],$$

where

$$Z_k = \log(I_{\boldsymbol{X}_N}(\lambda_k)) - \log(I_{\boldsymbol{Y}_N}(\lambda_k))$$

and $\hat{\mu}(\lambda_k)$ is the local maximum log-likelihood estimator of

$$\mu(\lambda_k) = \log(f_{\boldsymbol{X}_N}(\lambda_k)) - \log(f_{\boldsymbol{Y}_N}(\lambda_k))$$

computed by local linear fitting. The second distance is given by

$$d_{\mathrm{ISD}}(\boldsymbol{X}_N,\boldsymbol{Y}_N) = \int_{-\pi}^{\pi}(\hat{m}_{\boldsymbol{X}_N}(\lambda) - \hat{m}_{\boldsymbol{Y}_N}(\lambda))^2 d\lambda,$$

where $\hat{m}_{\boldsymbol{X}_N}(\lambda)$ and $\hat{m}_{\boldsymbol{Y}_N}(\lambda)$ are local linear smoothers of the log-periodograms obtained with the maximum local likelihood criterion.

Dissimilarity based on the symbolic representation SAX. The symbolic approximation representation (SAX) has been introduced by Lin et al. [68] and became one of the best symbolic representations for most time series problems [69]. The original data are first transformed into the Piecewise Aggregate Approximation (PAA) representation [34] — the signal is sliced into $w \geq 2$ segments of equal length and for each segment, the mean value is computed. Then, each segment is represented as a concatenation of $\alpha > 2$ symbols (letters of the alphabet). In this way, we obtain the SAX representation of a time series — $\hat{\boldsymbol{X}}_\alpha = (\hat{X}_1,\dots,\hat{X}_\alpha)$. For an alphabet $\boldsymbol{l}$, consisting of symbols l_i for $i = 1,\dots,\alpha$, we can define a distance between a pair of symbols l_i and l_j:

$$d_\alpha(l_i,l_j) = \begin{cases} 0 & \text{if } |i-j| \leq 1 \\ z_{(\max(i,j)-1)/\alpha} - z_{\min(i,j)/\alpha} & \text{otherwise} \end{cases},$$

where z_α is an α-quantile of the $N(0,1)$. Based on this, the dissimilarity between two SAX representations of time series may be defined as:

$$d_{\mathrm{SAX}}(\hat{\boldsymbol{X}}_\alpha,\hat{\boldsymbol{Y}}_\alpha) = \sqrt{\frac{N}{w}}\sqrt{\sum_{i=1}^{w}\left[d_\alpha(\hat{X}_i,\hat{Y}_i)\right]^2}.$$

For the full outline of dissimilarity measures based on SAX representation see [70]. Concerning the parameter choice, we followed [70], with the simplification that we set $\alpha = 10$ and $w = \frac{N}{3}$, where N is the length of the time series.

2.4. Structure-based distance measures

Structure-based distance measures try to find some higher-level structures i.e. to fit some model (e.g. ARMA) or to utilize the level of shared information by two time series (e.g. using the idea of the Kolmogorov complexity). However, it seems that – due to the wide range of meaning "structure" – the catalog remains open.

Piccolo distance. One of the first approaches applying structure-based outcome by fitting a model was proposed by Piccolo [71]. For the class of invertible ARIMA processes, denoting the vectors of $AR(k_1)$ and $AR(k_2)$ for $\boldsymbol{X}_N$ and $\boldsymbol{Y}_N$ respectively by

$$\hat{\boldsymbol{\Pi}}_{\boldsymbol{X}_N} = (\hat{\pi}_{1,\boldsymbol{X}_N},\dots,\hat{\pi}_{k_1,\boldsymbol{X}_N})$$

and

$$\hat{\boldsymbol{\Pi}}_{\boldsymbol{Y}_N} = (\hat{\pi}_{1,\boldsymbol{Y}_N},\dots,\hat{\pi}_{k_2,\boldsymbol{Y}_N}),$$

the following dissimilarity measure was proposed:

$$d_{\mathrm{PIC}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = \sqrt{\sum_{j=1}^{k} (\hat{\pi}'_{j,\boldsymbol{X}_N} - \hat{\pi}'_{j,\boldsymbol{Y}_N})^2},$$

where $k = \max(k_1, k_2)$, $\hat{\pi}'_{j,\boldsymbol{X}_N} = \hat{\pi}_{j,\boldsymbol{X}_N}$ if $j \le k_1$ and $\hat{\pi}'_{j,\boldsymbol{X}_N} = 0$ otherwise and analogously $\hat{\pi}'_{j,\boldsymbol{Y}_N} = \hat{\pi}_{j,\boldsymbol{Y}_N}$ if $j \le k_2$ and $\hat{\pi}'_{j,\boldsymbol{Y}_N} = 0$ otherwise. If the time series are non-stationary or they pose seasonality, they should be adjusted to satisfy the assumptions of the ARMA model.

Compression-based dissimilarity. Keogh et al. [72] proposed compression-based dissimilarity measure defined as

$$d_{\mathrm{CDM}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \frac{\mathrm{C}(\boldsymbol{X}_N, \boldsymbol{Y}_M)}{\mathrm{C}(\boldsymbol{X}_N)\mathrm{C}(\boldsymbol{Y}_M)}.$$

The CDM distance is descended from normalized compression distance (NCD) proposed by Lin et al. [68]:

$$d_{\mathrm{NCD}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = \mathrm{C}(\boldsymbol{X}_N, \boldsymbol{Y}_N) - \frac{\max\left(\mathrm{C}(\boldsymbol{X}_N), \mathrm{C}(\boldsymbol{Y}_M)\right)}{\min\left(\mathrm{C}(\boldsymbol{X}_N), \mathrm{C}(\boldsymbol{Y}_M)\right)},$$

where $\mathrm{C}(\boldsymbol{X}_N)$ denotes compression size of $\boldsymbol{X}_N$ and is used as an approximation of Kolmogorov complexity. $\mathrm{C}(\boldsymbol{X}_N)$ may be computed as the size of $\boldsymbol{X}_N$ compressed using data compressors for example: bzip2, gzip or xz. We used in calculations the best compression method (separately) for both series and concatenation. It is worth mentioning that CDM has recently achieved very good results in the field of natural language processing [73]. This is another argument in favor of developing simple distance measures, as they may have non-obvious applications.

Permutation distribution clustering. Dissimilarity measures based on permutation distribution clustering (PDC) use a permutation $\Pi(\boldsymbol{X}'_N)$ of m-dimensional embedding of $\boldsymbol{X}_N$. The dissimilarity between two time series $\boldsymbol{X}_N$ and $\boldsymbol{Y}_N$ is expressed in terms of the divergence between the distribution of these permutations, denoted by $P(\boldsymbol{X}_N), P(\boldsymbol{Y}_N)$. Specifically, Brandmaier [74] proposed the α-divergence between $P(\boldsymbol{X}_N)$ and $P(\boldsymbol{Y}_N)$ as a dissimilarity between time series $\boldsymbol{X}_N$ and $\boldsymbol{Y}_N$. The α-divergence (or the Rényi divergence) is a generalization of the Kullback–Leibler divergence.

2.5. Combined distance measures

Combined distance measures are convex combinations of some other distances. Mainly, they allow to capture different aspects of data and because of that, they can outperform their components. Parameters of combinations were selected from the range 0 to 1 (with step 0.01) using leave-one-out cross-validation. In this section, slightly different from the previous ones, we only provide main formulas, rather without wider comment, as component distances have already been discussed.

Derivative (combined) distance with ED (DD_{ED}). $\mathrm{DD}_{\mathrm{ED}}$ [75] is a distance measure defined

$$\mathrm{DD}_{\mathrm{ED}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = (1-\alpha)\,\mathrm{ED}(\boldsymbol{X}_N, \boldsymbol{Y}_N) + \alpha\,\mathrm{DED}(\boldsymbol{X}_N, \boldsymbol{Y}_N),$$

where α is a parameter $\alpha \in [0, 1]$.

Derivative (combined) distance with DTW (DD_{DTW}). $\mathrm{DD}_{\mathrm{DTW}}$ [75] is a distance measure defined

$$\mathrm{DD}_{\mathrm{DTW}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = (1-\alpha)\,\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \alpha\,\mathrm{DDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M),$$

where α is a parameter $\alpha \in [0, 1]$.

Second Derivative (combined) distance with DTW ($2DD_{DTW}$). $2\mathrm{DD}_{\mathrm{DTW}}$ [42] is a distance measure defined

$$\begin{aligned}2\mathrm{DD}_{\mathrm{DTW}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = {} & \alpha\,\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \\ & \beta\,\mathrm{DDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \\ & \gamma\,2\mathrm{DDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M),\end{aligned}$$

where α, β, γ are parameters, $\alpha + \beta + \gamma = 1$.

Transform distance with DTW (TD_{DTW}). $\mathrm{TD}_{\mathrm{DTW}}$ [76] is a distance measure defined

$$\mathrm{TD}^{T}_{\mathrm{DTW}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) = (1-\alpha)\,\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_N) + \alpha\,\mathrm{DTW}(T(\boldsymbol{X}_N), T(\boldsymbol{Y}_N)),$$

where α is a parameter $\alpha \in [0, 1]$ and T is a transform $T : R^N \to R^N$. There are three transforms used in [76]: cosine transform, sine transform, and Hilbert transform. All three are non-isometric for the DTW distance measure. For a series $\boldsymbol{X}_N$ we have a transform $T(\boldsymbol{X}_N) = (y_k : k = 1, 2, \ldots, N)$.

Cosine transform:

$$y_k = \sum_{i=1}^{N} x_i \cos\left[\frac{\pi}{N}\left(i - \frac{1}{2}\right)(k-1)\right].$$

Sine transform:

$$y_k = \sum_{i=1}^{N} x_i \sin\left[\frac{\pi}{N}\left(i - \frac{1}{2}\right)k\right].$$

Hilbert transform:

$$y_k = \sum_{\substack{i=1\\ i\neq k}}^{N} \frac{x_i}{k-i}.$$

We can use these transforms with DTW distance: $\mathrm{DTW}(T(\boldsymbol{X}_N), T(\boldsymbol{Y}_N))$. Regarding transform T is cosine (C), sine (S), or Hilbert (H) transform we have three distance measures:

CDTW, SDTW, HDTW.

Then we define three Transform Distances with DTW:

$$\begin{aligned}\mathrm{TD}^{C}_{\mathrm{DTW}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) &= (1-\alpha)\,\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_N) + \alpha\,\mathrm{CDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_N),\\ \mathrm{TD}^{S}_{\mathrm{DTW}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) &= (1-\alpha)\,\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_N) + \alpha\,\mathrm{SDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_N),\\ \mathrm{TD}^{H}_{\mathrm{DTW}}(\boldsymbol{X}_N, \boldsymbol{Y}_N) &= (1-\alpha)\,\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_N) + \alpha\,\mathrm{HDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_N),\end{aligned}$$

where α is a parameter $\alpha \in [0, 1]$.

Integral (combined) distance with DTW (ID_{DTW}). $\mathrm{ID}_{\mathrm{DTW}}$ [43] is a distance measure defined

$$\mathrm{ID}_{\mathrm{DTW}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = (1-\alpha)\,\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \alpha\,\mathrm{IDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M),$$

where α is a parameter $\alpha \in [0, 1]$.

Derivative (combined) distance with LCSS (DD_{LCSS}). $\mathrm{DD}_{\mathrm{LCSS}}$ [52] is a distance measure defined

$$\mathrm{DD}_{\mathrm{LCSS}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = (1-\alpha)\,\mathrm{LCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \alpha\,\mathrm{DLCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M),$$

where α is a parameter $\alpha \in [0, 1]$.

Second Derivative (combined) distance with LCSS ($2DD_{LCSS}$). $2\mathrm{DD}_{\mathrm{LCSS}}$ [52] is a distance measure defined

$$\begin{aligned}2\mathrm{DD}_{\mathrm{LCSS}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = {} & \alpha\,\mathrm{LCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \\ & \beta\,\mathrm{DLCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \\ & \gamma\,2\mathrm{DLCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M),\end{aligned}$$

where α, β, γ are parameters, $\alpha + \beta + \gamma = 1$.

Distance (combined) with DTW and LCSS (DD^{LCSS}_{DTW}). $\mathrm{DD}^{\mathrm{LCSS}}_{\mathrm{DTW}}$ [50] is a distance measure defined

$$\mathrm{D}^{\mathrm{LCSS}}_{\mathrm{DTW}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = (1-\alpha)\,\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \alpha\,\mathrm{LCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M),$$

where α is a parameter $\alpha \in [0, 1]$.

Derivative (combined) distance with DTW and LCSS (DD^{LCSS}_{DTW}). $\mathrm{DD}^{\mathrm{LCSS}}_{\mathrm{DTW}}$ [50] is a distance measure defined

$$\begin{aligned}\mathrm{DD}^{\mathrm{LCSS}}_{\mathrm{DTW}}(\boldsymbol{X}_N, \boldsymbol{Y}_M) = {} & \alpha\,\mathrm{DTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \\ & \beta\,\mathrm{LCSS}(\boldsymbol{X}_N, \boldsymbol{Y}_M) + \\ & \gamma\,\mathrm{DDTW}(\boldsymbol{X}_N, \boldsymbol{Y}_M),\end{aligned}$$

where α, β, γ are parameters, $\alpha + \beta + \gamma = 1$.

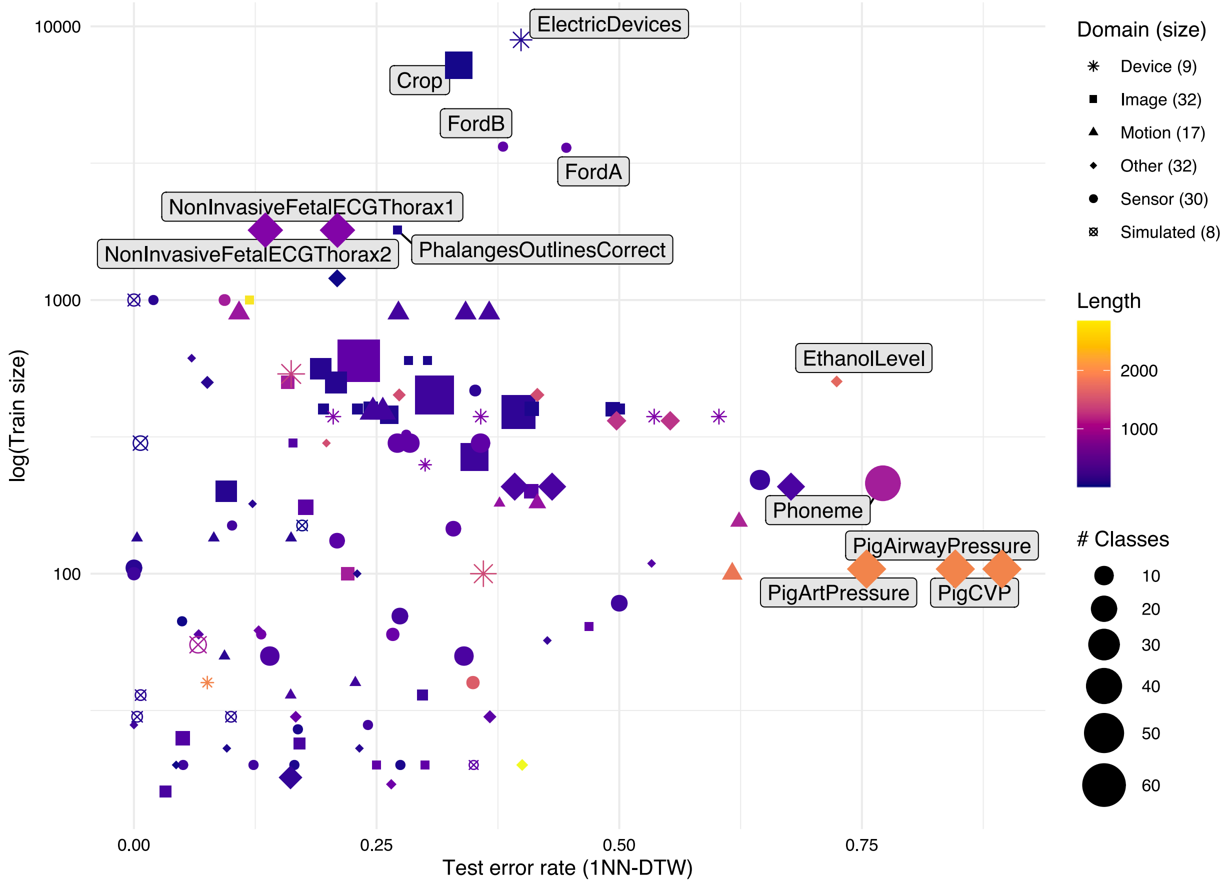

Fig. 2. Graphical summary of datasets in the UCR archive.

3. Experimental setup

We performed a comparison of 56 distances on all 128 datasets from the UCR time series archive [18]. A summary of all distances (with used parameters values) utilized in this paper is given in Table 3. The datasets originate from a plethora (15) of different domains. Within the data, the number of classes ranges from 2 to 60, the number of time series per dataset ranges from 40 to 24 000 samples, and time series lengths range from 15 to 2 844. In the latest version of the UCR database, there exist several datasets with missing observations and uneven sample lengths. However, the repository includes a standardized version of the database without these impediments, and that is the version we used. Also, before computations, data were z-normalized. In Fig. 2 we can observe some information about datasets. Additionally, we can observe how hard every dataset is to classify. Detailed information about datasets can be found on the website devoted to the UCR archive.

All datasets are split into a training and testing subset, and all parameter optimization is conducted on the training set only. We employ a train/test split for a few key reasons. First, it is a standard approach when working with the UCR archive. Second, this method is used for some datasets to ensure the train/test split minimizes bias [77,78]. The creators of the UCR [79] illustrate this with an example related to detecting alcohol fraud. In their experiments, a variety of bottles were utilized. They meticulously ensured that data from the same bottle were not included in both the training and testing datasets. This precaution was taken to avoid the algorithm mistakenly identifying variations in the bottles themselves, instead of the intended focus on detecting disparities in alcohol levels. Combining to perform a cross-validation would reintroduce the bias. And finally, but most importantly, it is not computationally feasible to cross-validate everything.

For statistical hypothesis testing purposes, we are assuming that these datasets are a random sample from the set of all possible time series classification problems. This is true in the sense that these data were not collected with any agenda in mind and are not more suitable for any particular algorithm.

In our paper, we use the methodology proposed by Keogh and Kasetty [80], hence we evaluate the efficacy of distance measures using the classification accuracy of the 1NN classifier. While one should be aware that the proposed technique cannot present the overall evaluation of a distance measure, there seem to be more advantages than disadvantages of the chosen method. For example, Wang et al. [19] pointed out three aspects: simplicity of implementation, performance directly dependent on distance choice, and relatively (to other, often more complex classifiers) good quality of classification.

As the aim of our research was to compare as many distances as possible, we tried to utilize as much code as possible that already exists. Thus, one part of the calculations was done using the R software [81] and the following libraries: `TSclust` [32], `TSdist` [41], `tsmp` [82]. The second part of the distances (especially the computationally demanding ones), were computed using our implementations in C++. The computations were carried out using Intel Xeon E5-2697 architecture provided by Poznań Supercomputing and Networking Center. For the analysis, we used the `scmamp` [83] library with our modifications.

4. Results

The error rates on the test subset with the 1NN classifier are shown for each measure in Tables 4 and 5. Fig. 3 demonstrates the comparison of error rates and ranks for all distances. These results lead to a conclusion that even though there is no clear winner, the top distance measures are dominated by DTW methods — both modifications of

Table 3
Summary of all used distances. EL (Equal Length) column denotes if a distance requires a time series of equal length. For each parameter, the values that were used are provided. If they are not specified, it means that they were optimized (unless otherwise indicated in the text, the method used was leave-one-out CV (LOOCV).

Abbreviation	Distance's name	Category	Parameters	EL	Metrics
$2DD_{DTW}$	Second Derivative distance with DTW	Combined	α, β, γ	No	No
$2DD_{LCSS}$	Second Derivative distance with LCSS	Combined	$\alpha, \beta, \gamma, \delta = 100\%, \varepsilon$	No	No
2DDTW	Second Derivative DTW	Shape-based	–	No	No
2DLCSS	Second Derivative LCSS	Edit-based	$\delta = 100\%,\ \varepsilon$	No	No
ACF	Autocorrelation-based	Feature-based	$L = 50,\ \boldsymbol{\Omega} = \boldsymbol{I}$	No	No
CC	Cross-correlation-based	Feature-based	–	Yes	No
CDM	Compression-based	Structure-based	–	No	No
CDTW	Cosine DTW	Combined	–	Yes	No
CID	Complexity-Invariant	Shape-based	–	Yes	No
CORT	Temporal Correlation and Raw Value Behaviors	Feature-based	$k = 2$	Yes	No
D_{DTW}^{LCSS}	Distance with DTW and LCSS	Combined	$\alpha, \delta = 100\%, \varepsilon$	No	No
DD_{DTW}^{LCSS}	Derivative distance with DTW and LCSS	Combined	$\alpha, \beta, \gamma, \delta = 100\%, \varepsilon$	No	No
DD_{DTW}	Derivative distance with DTW	Combined	α	No	No
DD_{ED}	Derivative distance with ED	Combined	α	Yes	No
DD_{LCSS}	Derivative distance with LCSS	Combined	$\alpha, \delta = 100\%, \varepsilon$	No	No
DDTW	Derivative DTW	Shape-based	–	No	No
DED	Derivative ED	Shape-based	–	Yes	No
DLCSS	Derivative LCSS	Edit-based	$\delta = 100\%,\ \varepsilon$	No	No
DTW	Dynamic Time Warping	Shape-based	–	No	No
DTWc	DTW with Sakoe-Chiba Band	Shape-based	w	No	No
ED	Euclidean	Shape-based	–	Yes	Yes
EDR	Edit Distance with Real Penalty	Edit-based	$\varepsilon = 0.1,\ \sigma = 25\%$	No	Yes
ERP	Edit Distance on Real Sequence	Edit-based	$g = 0.1,\ \sigma = 25\%$	No	No
FC	Fourier-coefficients-based	Feature-based	$n = \lfloor \frac{N}{2} \rfloor + 1$	Yes	No
GLK	Based on the Generalized Likelihood Ratio Test	Feature-based	–	Yes	No
HDTW	Hilbert DTW	Combined	–	Yes	No
ID_{DTW}	Integral distance with DTW	Combined	α	No	No
IDTW	Integral DTW	Shape-based	–	No	No
INF	Chebyshev norm	Shape-based	–	Yes	Yes
IP	Integrated periodogram	Feature-based	–	Yes	No
ISD	Based on the Integrated Squared Difference between the Log-Spectra	Feature-based	–	Yes	No
LCSS	Longest Common Subsequence	Edit-based	$\delta = 100\%,\ \varepsilon$	No	No
LLR_DLS	Based on non-parametric spectral estimator (least squares)	Feature-based	α	Yes	No
LLR_LK	Based on non-parametric spectral estimator (maximum likelihood)	Feature-based	α	Yes	No
LNP	Logarithmed-normalized-periodogram-based	Feature-based	–	Yes	No
MAN	Manhattan	Shape-based	–	Yes	Yes
MIN	Minkowski	Shape-based	$p = 3$	Yes	Yes
MPDist	Matrix Profile	Shape-based	$k = 5\% \cdot 2 \cdot N,\ L$	No	No
MSM	Move-Split-Merge	Edit-based	$c = 0.1$	No	Yes
NCD	Normalized-compression-based	Structure-based	–	No	No
NP	Normalized-periodogram-based	Feature-based	–	Yes	No
P	Periodogram-based	Feature-based	–	Yes	No
PACF	Partial-autocorrelation-based	Feature-based	$L = 50,\ \boldsymbol{\Omega} = \boldsymbol{I}$	No	No
PC2	Pearson's correlation based	Feature-based	$\beta = 1$	Yes	No
PDC	Permutation distribution clustering	Structure-based	–	Yes	No
PIC	Piccolo	Structure-based	–	No	No
SAX	Symbolic Aggregate Approximation	Feature-based	$\alpha = 10,\ w = \frac{N}{3}$	No	No
SDTW	Sine DTW	Combined	–	Yes	No
$shapeDTW_{HOG1D}$	Shape Dynamic Time Warping	Shape-based	–	No	No
$shapeDTW_{RawSub}$	Shape Dynamic Time Warping	Shape-based	–	No	No
STS	Short Time Series	Shape-based	–	Yes	No
TAM	Time Alignment Measurement	Shape-based	–	No	No
TD_{DTW}^{C}	Cosine Distance with DTW	Combined	α	Yes	No
TD_{DTW}^{H}	Hilbert Distance with DTW	Combined	α	Yes	No
TD_{DTW}^{S}	Sine Distance with DTW	Combined	α	Yes	No
TQ	TQuest	Feature-based	τ	No	No

DTW ($shapeDTW_{HOG1D}$, $shapeDTW_{RawSub}$, DTWc) and combinations of DTW (e.g. DD_{DTW}^{LCSS}, DD_{DTW}, $2DD_{DTW}$). Among the best non-DTW-based methods a good performance of ERP and MPdist should be highlighted, with median errors in the top 10. It is interesting to note that all 4 considered structure-based distances are positioned at the very end of the ranking, both taking into account median errors and median ranks. Also, looking at feature-based distances, one could observe rather moderate performance (measures placed mostly in the second half of the ranking).

We now want to compare selected methods in pairs. Of course, it is not possible to present all such comparisons for pairs of measures, as there are $\binom{56}{2} = 1540$ of them. For this reason, we selected the most interesting ones, taking into account the popularity of the methods and the results obtained. Fig. 4 shows scatter plots of errors for pairs of distances. Planned comparisons revealed that:

- slightly superior results are achieved by $shapeDTW_{HOG1D}$ in comparison with DD_{DTW}^{LCSS} and $2DD_{DTW}$, however $shapeDTW_{HOG1D}$ clearly outperforms DTW,
- combining information contained by DTW with its derivative significantly boosts performance (DD_{DTW} better than DTW and DDTW),
- adding second derivative of DTW to DD_{DTW} (and thus creating $2DD_{DTW}$) do not improve the efficiency while adding second derivative of LCSS to DD_{LCSS} (and thus creating $2DD_{LCSS}$) even worsen performance,

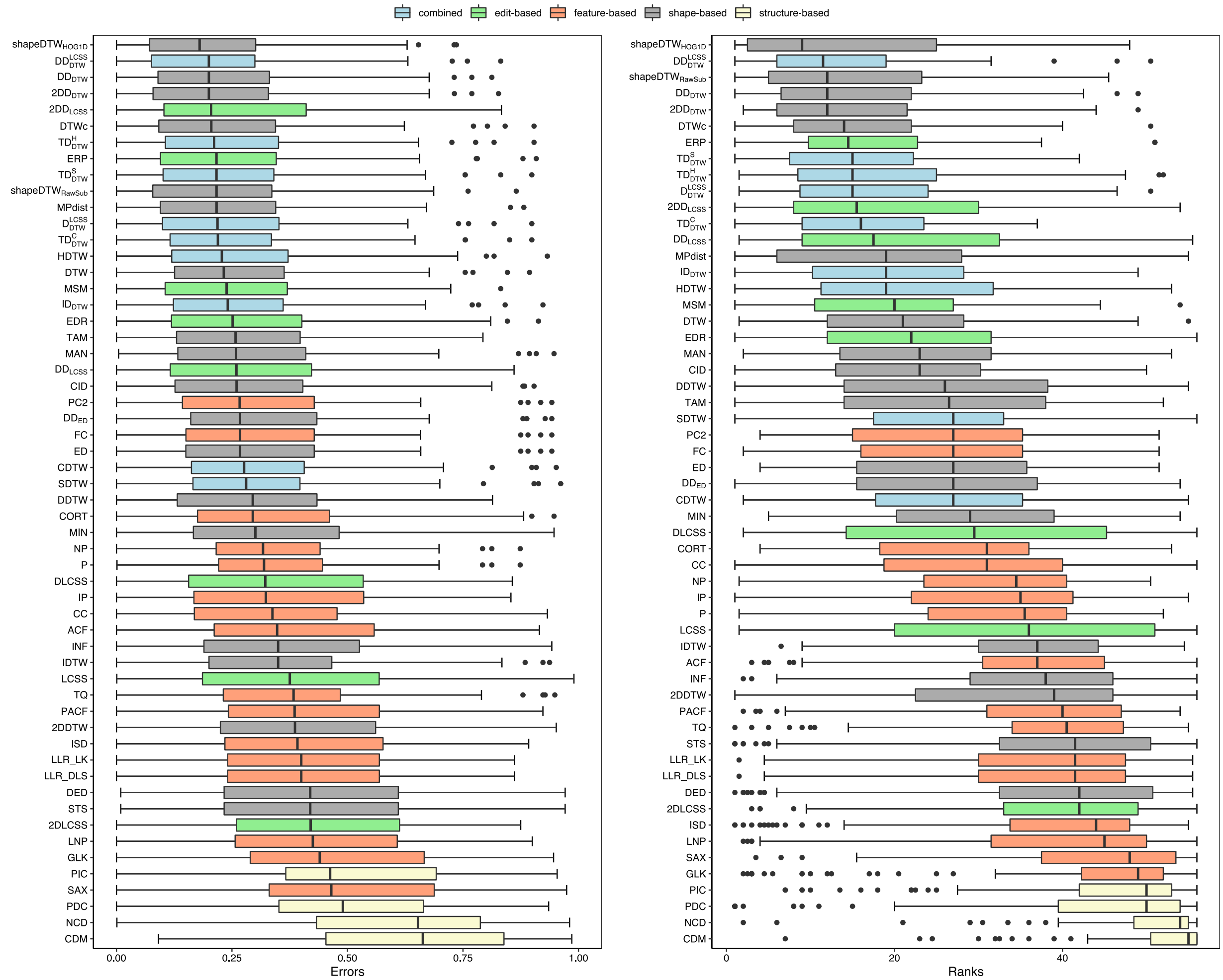

Fig. 3. Comparison of error rates and ranks.

- adding windowing to the warping path in DTW (and thus creating DTWc measure) improves not only the time complexity but also performance,
- ED seems to be significantly worse than DTW and $\text{shapeDTW}_{\text{HOG1D}}$, however, there are still over a dozen datasets where ED is a better choice.

Fig. 5 presents results of k-means clustering (we used mean errors and standard deviations as features). It seems that the reasonable number of clusters is equal to 3. In Fig. 6, mean errors and standard deviations of errors of each measure are shown. Additionally, measures are grouped according to clusters. These results suggest, that we may distinguish the 3 disjoint groups: weak and unstable (blue), moderate (red), and highly efficient and stable (green).

4.1. Statistical comparison of the used method

To identify differences between the methods, we present a detailed statistical comparison. At the beginning, we test the null hypothesis that all classifiers perform the same and the observed differences are merely random (omnibus test). The alternative states that at least one classifier is statistically different in terms of performance from the others. The Friedman test [84,85] with Iman and Davenport [86] extension is probably the most popular omnibus test, and it is usually a good choice when comparing more than five different algorithms [87,88]. Let R_{ij} be the rank of the jth of K classifiers on the ith of N datasets and

$$R_j = \frac{1}{N}\sum_{i=1}^{N} R_{ij}.$$

The test compares the mean ranks of classifiers and is based on the statistic

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)-\chi_F^2},$$

where

$$\chi_F^2 = \frac{12N}{K(K+1)}\sum_{i=1}^{K} R_i^2 - 3N(K+1)$$

is the Friedman statistic, which has the F distribution with $K-1$ and $(K-1)(N-1)$ degrees of freedom. The p-value from this test is equal to 0 ($F_F = 68$). The obtained p-value indicates that we can safely reject the null hypothesis that all the algorithms perform the same. We can therefore try to detect significant pairwise differences among all of the classifiers. For that, we have two possibilities: post-hoc tests and pairwise tests.

4.1.1. Nemenyi's post-hoc test

Demšar [89] proposes the use of the Nemenyi's test [90] that compares all the algorithms pair-wise. For a significance level α the test determines the critical difference (CD). If the difference between the average ranking of two algorithms is greater than

$$\text{CD} = q_\alpha\sqrt{\frac{K(K+1)}{6N}}$$

the null hypothesis that the algorithms have the same performance is rejected (q_α are based on the Studentized range statistic divided by $\sqrt{2}$). Demšar [89] proposes a plot to visually check the differences, the

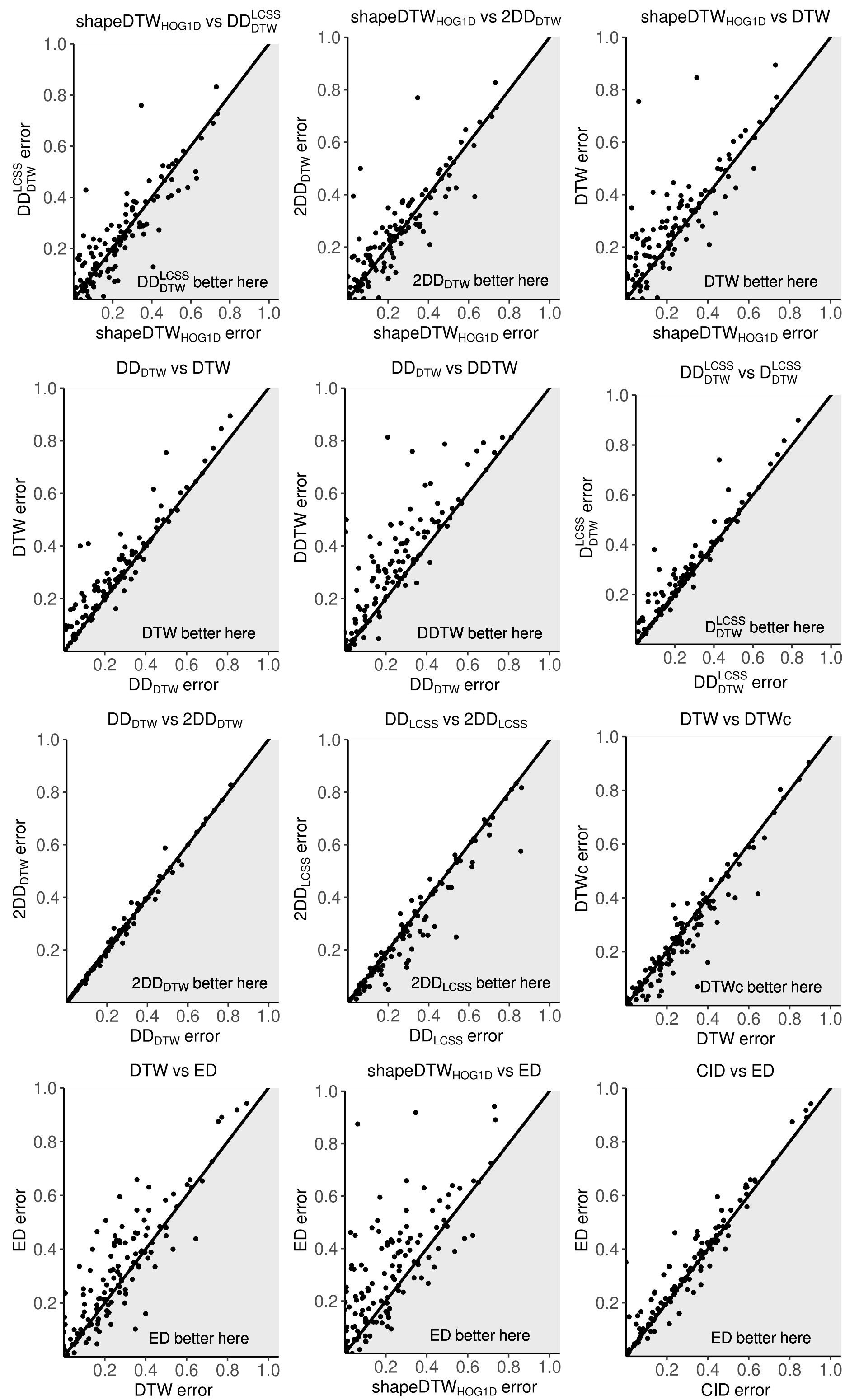

Fig. 4. Comparison of error rates.

CD plot. In the plot, those algorithms that are not joined by a line can be regarded as different.

In our case, with a significance of $\alpha = 0.05$, any two algorithms with a difference in the mean rank above 8.7038 will be regarded as non-equal (Fig. 7).

4.1.2. Corrected pairwise tests

The Nemenyi's test is the most simple but less powerful alternative, hence we can use the second approach i.e. using a classical test to assess all the pair-wise differences between algorithms and then correct the p-values for multiple testing. In our case we should use the

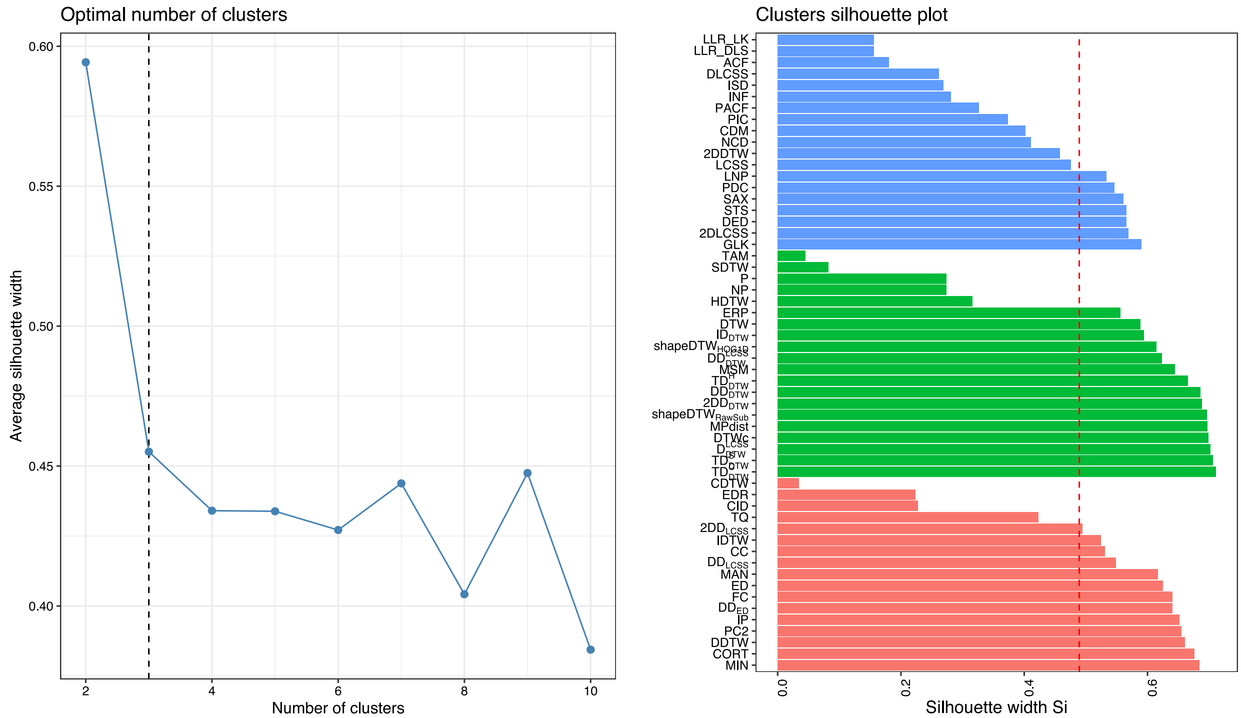

Fig. 5. Results of clustering using k-means algorithm.

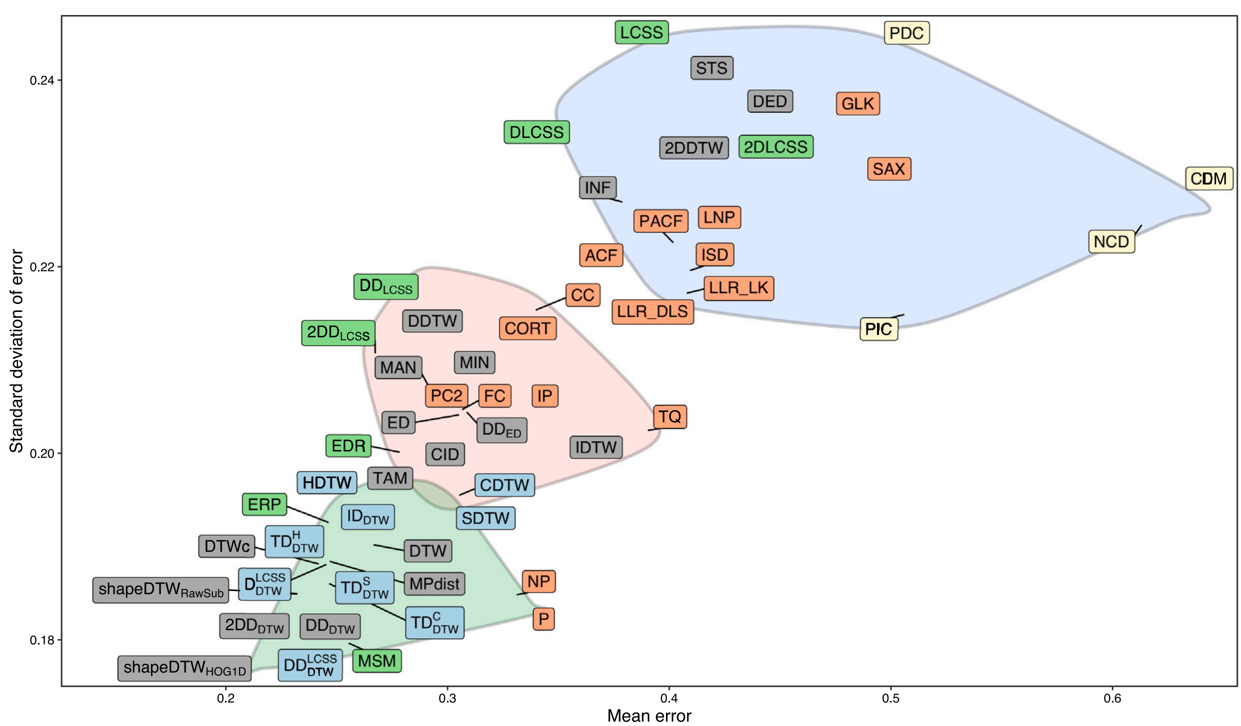

Fig. 6. Mean errors and standard deviations of error for each measure. The color coding of the boxes corresponds to the measure categories, consistent with Fig. 3. Measures are organized into three distinct clusters determined by the k-means clustering algorithm.

Wilcoxon signed-ranks test [91] which is a non-parametric alternative to the paired t-test. The paired t-test is used to compare the means of two related groups to determine if there is a statistically significant difference between them. The null hypothesis in the Wilcoxon signed-ranks test states that there is no difference between the two classifiers. The alternative hypothesis states that there is a difference between the two classifiers.

Let d_i be the difference between the performance scores of the two classifiers on the ith dataset. The differences are ranked according to their absolute values; average ranks are assigned in case of ties. Let

$$R^+ = \sum_{d_i>0}(d_i) + \frac{1}{2}\sum_{d_i=0}(d_i),$$

$$R^- = \sum_{d_i<0}(d_i) + \frac{1}{2}\sum_{d_i=0}(d_i),$$

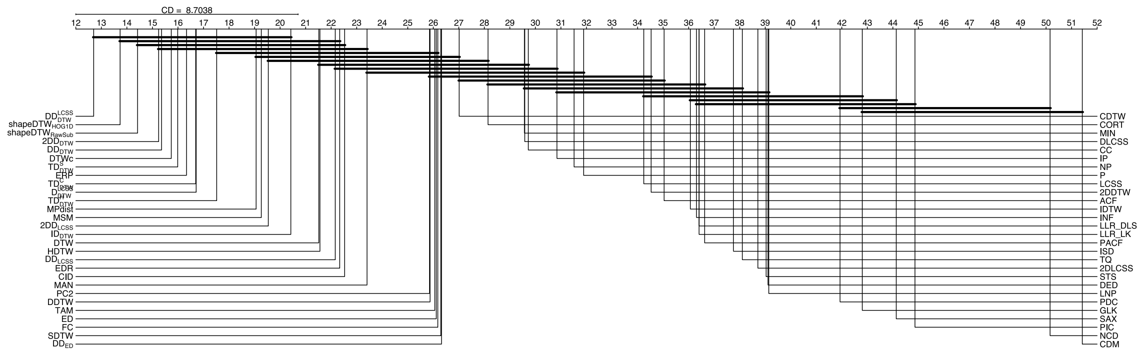

Fig. 7. Critical difference plot.

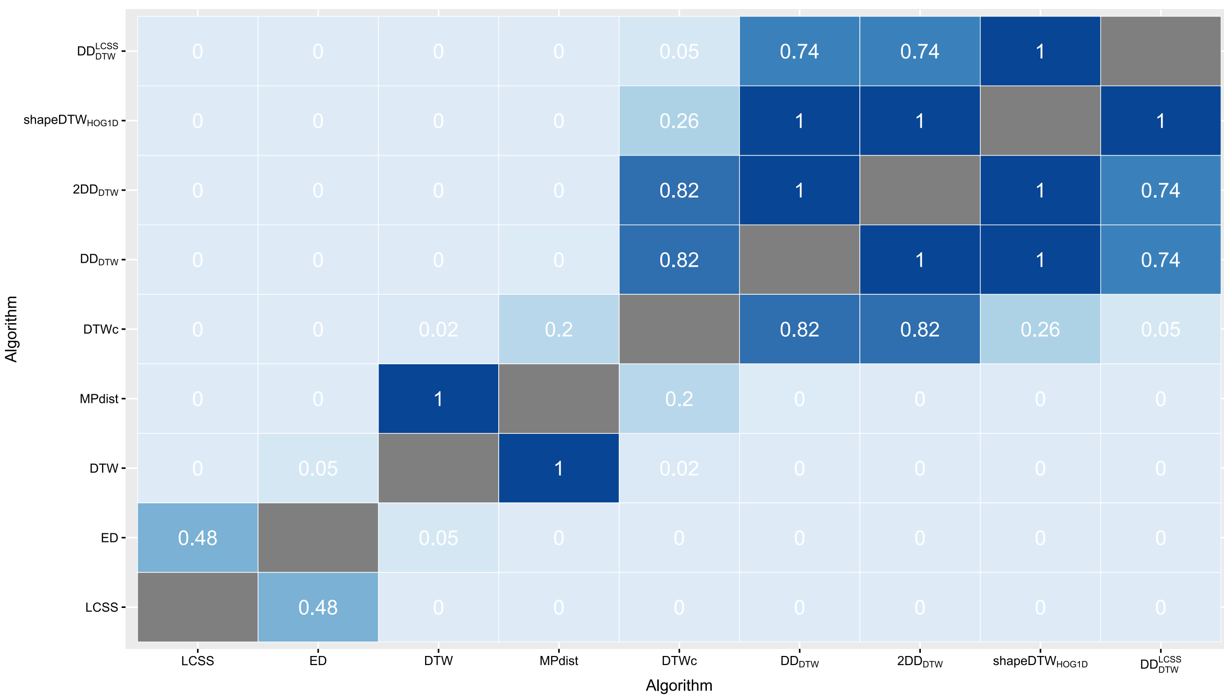

Fig. 8. Corrected p-values using Bergmann and Hommel's procedure.

$W = \min(R^+, R^-)$.

Under the null hypothesis, W follows a specific distribution with no simple expression. Critical W values by values of N can be found in statistics textbooks. For large samples with $N > 25$ the W statistics approximates normal distribution and we reject null hypothesis if

$$\frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} < z_{\alpha/2},$$

where $z_{\alpha/2}$ is a quantile from standard normal distribution. To retain an overall significance level α, one has to adjust the value of α for each post-hoc comparison. There are various methods for this. García and Herrera [87] explain and compare the use of various correction algorithms and recommend Bergmann and Hommel's method [92]. They show that it is the most powerful. Bergmann and Hommel's correction is a costly method — in computational terms. Comparing more than nine algorithms with the Bergmann and Hommel procedure is unfeasible in practice [83]. Hence, we selected only 9 algorithms to compare statistically using this method (this required checking 54 466 sets of hypotheses). We selected algorithms considering the quality of results and popularity.

We can directly visualize the corrected p-value matrix generated when doing all the pairwise comparisons (Fig. 8).

Conversely, to what happened with Nemenyi's test, it makes no sense to draw a critical difference plot, since the critical differences are not constant throughout the comparisons. However, we can plot a graph (Fig. 9) where the algorithms are the nodes and two nodes are linked if the null hypothesis of being equal cannot be rejected at a significance level $\alpha = 0.05$. This is a visualization for the matrix of adjusted p-values (Fig. 8), from which reading all dependencies is more difficult and much more time-consuming.

Figs. 8 and 9 lead to a conclusion, that two groups can be distinguished among the selected 9 methods. The leading methods are statistically insignificantly different from each other: $\mathrm{DD}_{\mathrm{DTW}}^{\mathrm{LCSS}}$, $\mathrm{shapeDTW}_{\mathrm{LCSS}}$, $2\mathrm{DD}_{\mathrm{DTW}}$, $\mathrm{DD}_{\mathrm{DTW}}$, DTWc. The second group – methods statistically significantly worse than almost any other from the first group and of similar efficiency within it – may be formed from MPdist, DTW, ED, and LCSS. A further, more specific findings are:

- comparing DTW with DTW-based methods (included in the selected 9 methods) shows the significant advantage of the latter ones,
- referring to $\mathrm{DD}_{\mathrm{DTW}}$, adding the second derivative (thus creating $2\mathrm{DD}_{\mathrm{DTW}}$) or LCSS component (thus creating $\mathrm{DD}_{\mathrm{DTW}}^{\mathrm{LCSS}}$) does not differentiate methods statistically, however it enhance the average rank,

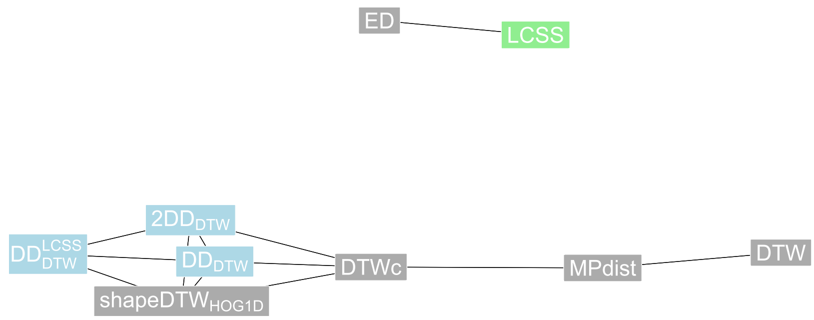

Fig. 9. Graph of methods for Bergmann and Hommel's procedure. Boxes are colored according to the category of a measure, same as in Fig. 3.

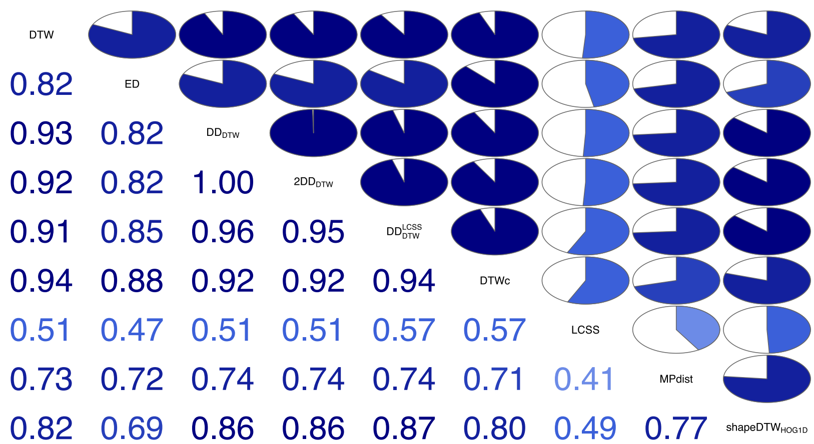

Fig. 10. Spearman's rank correlation coefficient of examined methods (the areas of the circles are proportional to Spearman's rank correlation coefficients).

- there is no statistically significant difference between MPdist and both DTW and DTWc.

4.2. Correlation analysis

Fig. 10 presents Spearman's rank correlation coefficient for 9 selected methods. The Spearman's correlation coefficient is a nonparametric statistic for evaluating the strength of association between two variables [93]. Compared to Pearson's correlation coefficient, Spearman's correlation coefficient operates on the ranks of the data rather than the raw data. It can be calculated using the following equation:

$$r_S = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{N(N^2-1)},$$

where d_i is the difference between ranks for each dataset, and N is the number of datasets.

The first and the most important conclusion we can draw from Fig. 10 is that all pairs of selected 9 distances are positively correlated, so all of them show similar behavior on the same datasets (higher accuracy for the same easy datasets and lower for the same harder ones). We may observe that LCSS and MPdist are seemingly less correlated with all other distances than the other ones. That may lead to a conclusion that these distances may be perceived as complementary to the others.

5. Conclusions and future work

In this paper, we have provided an extensive comparison, supported by comprehensive statistical analysis, which is – to the best of our knowledge – the biggest available study taking into account both the number of datasets (128) and the number of distances (56). We followed the distances' categorization proposed by Esling and Agón [33] but we extended it by one group — combined distances. They are distances proposed in the last few years and may be defined as distances made of a convex combination of some other distances.

Overall, our results indicate that distances based on DTW dominated the comparison (8 representatives in the top 10 average error rates). However, there is no clear-cut winner. It is hard to point out a single metric to look at – average rank, mean error, or number of wins – just to name a few. Thus, we tried to provide an extensive statistical analysis revealing different aspects of compared measures. We performed clustering (taking into account mean error and standard deviation), designating 3 disjoint groups of distance measures: weak and unstable, moderate and highly efficient and stable. Also, a statistical analysis using Bergmann and Hommel's procedure was done, and as a result leading group – statistically insignificantly different between each other – was specified: $\mathrm{DD}_{\mathrm{DTW}}^{\mathrm{LCSS}}$, $\mathrm{shapeDTW}_{\mathrm{LCSS}}$, $2\mathrm{DD}_{\mathrm{DTW}}$, $\mathrm{DD}_{\mathrm{DTW}}$, DTWc. Nevertheless, the analysis showed that MPdist may be considered as good as DTWc.

Several more general conclusions may be drawn. It seems, that using information originating from different sources and thus creating combined distances is in general beneficial and may be a good direction to follow. Also, in most cases of combined measures adding a derivative significantly boosts performance but it is not always the rule.

Distance-based methods in time series classification, while popular and effective in many scenarios, do have certain limitations compared to other approaches. Here are some of the key limitations:

- Sensitivity to noise and outliers: distance-based methods, especially those using measures like Euclidean distance, can be sensitive to noise and outliers in the data.
- Need for predefined distance measures: distance-based methods rely on a predefined distance measure, which may not capture the true nature of the similarity between time series. Choosing an appropriate distance measure is crucial, and the wrong choice can significantly impact performance.

- Difficulty with multivariate time series: while distance-based methods are effective for univariate time series, they can be less effective or more complex to implement for multivariate time series, where the interaction between different variables needs to be considered.
- Dependency on length of time series: the performance of distance-based methods can depend on the length of the time series (details in Table 3, column EL). Inconsistent lengths between training and testing data can pose challenges.
- Limitation in capturing complex patterns: basic distance measures may not effectively capture more complex patterns and relationships in time series data, such as non-linear relationships.
- Inefficiency in high-dimensional spaces: distance-based methods can suffer from the curse of dimensionality, where the performance degrades as the number of dimensions increases.
- Feature extraction dependency: for some distance measures, effective feature extraction is essential. If the feature extraction process is not robust, the overall classification performance can suffer.
- Lack of model flexibility: compared to model-based approaches or deep learning methods, distance-based methods can be less flexible in learning from data, as they do not inherently model the underlying process generating the data.

Despite these limitations, distance-based methods remain a popular choice due to their versatility, simplicity, ease of implementation, and ability to work with raw data (these methods can often work effectively with raw time series data, without the need for extensive feature engineering or preprocessing), effectiveness in many scenarios (especially with small data sets), and the extensive research supporting their use. However, for certain applications, other methods such as deep learning or probabilistic modeling might be more appropriate.

We supplied a full table of results. We hope that the study will be useful for machine learning practitioners as well as researchers aiming to discover new distance measures.

We freely acknowledge there are several limitations of our study, which may be developed in the future:

1. Among others, it would be interesting to check how the conclusions made in this work reflect other time series mining tasks involving computing distance measures e.g. clustering or anomaly detection.
2. Additionally, the analysis of the distributions of distances within and between classes could be done. The idea is to use this approach for a more nuanced comparison, potentially employing metrics like AUC, which is indeed valuable and could offer additional depth to the study.
3. Combining the training and test sets into one dataset and selecting all parameters through cross-validation could also be interesting.
4. Finally, the paper could also be expanded to include much more computationally intensive methods such as ROCKET and similar distance measures.

6. Additional tables

See Tables 4 and 5.

Table 4

Error rates (in %) of all considered distance measures using the 1NN classifier. The best-performing classifier for each dataset is highlighted in bold. The final six rows show the number of wins per distance, average ranks, mean and median error rates, the standard deviation of the error rates, and the maximum error rate.

Dataset	$2DD_{DTW}$	$2DD_{LCSS}$	2DDTW	2DLCSS	ACF	CC	CDM	CDTW	CID	CORT	D^{LCSS}_{DTW}	DD^{LCSS}_{DTW}	DD_{DTW}	DD_{ED}	DD_{LCSS}	DDTW	DED	DLCSS	DTW	DTWc	ED	EDR	ERP	FC	GLK	HDTW	ID_{DTW}	IDTW
ACSF1	37.0	81.0	41.0	81.0	31.0	48.0	75.0	32.0	46.0	46.0	36.0	37.0	37.0	46.0	81.0	37.0	46.0	81.0	36.0	38.0	46.0	33.0	**16.0**	47.0	42.0	39.0	37.0	66.0
Adiac	29.9	57.5	62.4	66.2	44.0	38.1	92.6	38.6	37.3	39.6	39.6	30.4	29.9	37.6	85.7	41.2	41.9	85.7	39.6	39.1	38.9	58.6	37.9	38.9	81.1	44.5	39.4	64.2
AllGestureWiimoteX	27.4	32.6	57.9	67.4	55.6	62.4	88.4	47.6	47.4	53.0	**25.0**	**25.0**	27.3	48.4	38.7	40.9	69.9	55.4	28.4	28.3	48.4	50.4	31.0	48.4	75.0	31.6	28.4	47.7
AllGestureWiimoteY	25.3	28.7	56.6	76.3	56.0	50.1	86.4	42.4	44.9	47.1	24.7	24.7	25.3	45.4	28.6	37.9	70.4	54.0	27.1	27.0	43.1	42.0	30.3	43.1	81.1	29.6	26.6	46.0
AllGestureWiimoteZ	35.7	42.6	61.4	77.7	57.9	61.1	87.1	48.0	51.3	59.1	35.3	35.0	35.7	54.7	43.3	44.7	78.0	54.3	35.7	34.9	54.6	49.3	44.0	54.6	76.9	37.3	34.7	52.4
ArrowHead	21.1	18.9	62.9	42.3	28.6	19.4	68.6	30.3	**17.1**	21.7	28.0	19.4	21.1	20.0	18.9	28.0	40.6	18.9	29.7	20.0	20.0	25.1	22.9	20.0	40.6	37.1	25.7	38.3
Beef	30.0	30.0	36.7	40.0	56.7	36.7	76.7	33.3	36.7	33.3	36.7	33.3	33.3	33.3	30.0	33.3	30.0	30.0	36.7	33.3	33.3	30.0	40.0	33.3	63.3	33.3	36.7	50.0
BeetleFly	35.0	**5.0**	25.0	40.0	40.0	35.0	50.0	50.0	30.0	35.0	30.0	20.0	35.0	40.0	20.0	35.0	35.0	20.0	30.0	30.0	25.0	30.0	30.0	25.0	55.0	30.0	30.0	40.0
BirdChicken	15.0	**0.0**	40.0	15.0	15.0	40.0	45.0	35.0	35.0	45.0	25.0	25.0	15.0	40.0	**0.0**	20.0	45.0	**0.0**	25.0	30.0	45.0	10.0	20.0	45.0	5.0	20.0	25.0	35.0
BME	0.7	4.7	**0.0**	2.7	43.3	35.3	49.3	14.7	7.3	18.7	3.3	3.3	0.7	16.7	8.0	**0.0**	25.3	4.0	10.0	2.0	16.7	20.0	10.0	16.7	43.3	0.7	2.7	14.7
Car	20.0	16.7	45.0	31.7	61.7	26.7	81.7	35.0	26.7	25.0	26.7	23.3	20.0	26.7	18.3	30.0	33.3	16.7	26.7	23.3	26.7	11.7	21.7	26.7	51.7	31.7	31.7	41.7
CBF	0.3	1.2	53.2	40.7	31.6	25.9	41.1	2.6	1.6	20.6	0.3	0.3	0.3	14.8	1.2	45.3	66.0	39.8	0.3	0.4	14.8	6.0	0.2	15.1	44.9	**0.0**	0.3	1.1
Chinatown	5.2	14.8	15.4	22.9	14.0	4.4	39.1	4.1	4.1	3.8	4.3	5.2	5.2	4.6	29.0	10.7	10.1	28.7	4.3	4.7	4.7	3.8	3.2	4.7	22.2	8.1	4.1	4.3
ChlorineConcentration	27.8	43.9	**26.9**	44.1	42.2	35.2	52.8	35.2	35.1	34.8	35.2	29.0	29.2	27.7	49.8	28.7	27.7	49.3	35.2	35.0	35.0	36.8	37.4	35.1	57.6	37.9	35.8	44.7
CinCECGTorso	27.8	7.1	34.9	19.4	48.6	12.5	58.0	14.8	8.4	9.3	7.1	7.1	27.5	9.9	7.1	27.5	21.4	2.5	34.9	7.0	10.3	18.6	4.1	10.3	12.9	30.8	34.9	43.6
Coffee	**0.0**	7.1	21.4	14.3	7.1	**0.0**	46.4	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	7.1	7.1	7.1	7.1	**0.0**	**0.0**	**0.0**	3.6	7.1	**0.0**	25.0	**0.0**	**0.0**	14.3
DiatomSizeReduction	**3.3**	11.8	10.5	11.4	8.2	6.9	69.0	8.5	6.5	6.2	**3.3**	**3.3**	**3.3**	6.5	11.8	6.9	11.1	11.8	**3.3**	6.5	6.5	4.2	5.9	6.5	25.8	5.6	**3.3**	13.7
DistalPhalanxOutlineAgeGroup	29.5	38.9	36.0	41.7	29.5	38.1	61.2	36.7	36.7	36.7	**23.0**	29.5	29.5	34.5	31.6	38.1	39.6	32.4	**23.0**	37.4	37.4	28.1	27.3	36.7	48.9	30.2	33.1	41.0
DistalPhalanxOutlineCorrect	26.8	27.2	30.8	29.7	26.4	28.3	42.8	27.5	28.6	27.9	28.3	26.4	26.8	25.7	27.5	27.5	25.4	27.2	28.3	27.5	28.3	31.2	26.8	28.3	36.6	25.0	28.6	34.8
DistalPhalanxTW	38.9	43.2	43.9	48.2	45.3	38.1	76.3	35.2	37.4	38.8	41.0	38.9	38.9	43.2	43.2	45.3	46.0	43.2	41.0	36.7	36.7	36.7	36.7	36.7	51.1	33.1	39.6	41.7
DodgerLoopDay	58.8	43.8	76.2	76.2	72.5	47.5	88.8	42.5	50.0	57.5	50.0	50.0	48.8	46.2	51.2	78.8	83.8	71.2	50.0	41.2	45.0	48.8	**31.2**	46.2	81.2	46.2	50.0	50.0
DodgerLoopGame	12.3	10.1	44.2	42.0	34.8	12.3	52.2	13.0	10.9	17.4	8.7	8.7	12.3	11.6	10.1	43.5	49.3	36.2	12.3	7.2	11.6	25.4	**6.5**	10.9	41.3	9.4	13.0	19.6
DodgerLoopWeekend	5.8	1.4	19.6	31.9	5.8	**0.7**	26.1	2.2	1.4	2.2	1.4	1.4	5.8	1.4	1.4	23.2	27.5	22.5	5.1	2.2	1.4	5.1	1.4	1.4	43.5	2.9	3.6	7.2
Earthquakes	25.9	25.9	25.9	29.5	33.1	**25.2**	31.7	26.6	30.9	33.8	28.1	29.5	29.5	27.3	25.9	29.5	69.8	25.9	28.1	27.3	28.8	30.2	28.8	28.8	41.7	30.2	29.5	30.9
ECG200	16.0	13.0	18.0	30.0	26.0	11.0	41.0	16.0	11.0	12.0	12.0	12.0	17.0	10.0	11.0	13.0	15.0	17.0	23.0	12.0	12.0	16.0	15.0	12.0	20.0	19.0	19.0	22.0
ECG5000	7.7	7.4	9.7	9.0	9.1	8.0	25.1	7.4	7.2	7.0	7.4	7.6	7.6	7.6	**6.6**	7.7	7.7	7.0	7.6	7.5	7.5	7.3	6.7	7.5	17.7	7.8	7.6	10.3
ECGFiveDays	23.1	5.6	37.3	39.1	0.6	17.2	49.9	17.4	21.8	23.8	14.4	14.4	23.1	20.3	5.6	34.3	38.3	21.0	23.2	20.3	20.3	23.2	19.3	20.2	27.6	20.3	21.2	27.3
FaceAll	10.4	20.2	23.1	16.3	33.8	37.1	91.0	36.6	26.9	21.5	21.4	21.4	**9.8**	28.6	22.4	13.4	28.5	20.2	19.2	19.2	28.6	24.4	20.2	28.6	58.6	23.3	19.3	24.4
FaceFour	17.1	18.2	38.6	33.0	46.6	30.7	84.1	35.2	19.3	22.7	18.2	18.2	17.1	21.6	18.2	40.9	47.7	13.6	17.1	11.4	21.6	**4.5**	10.2	21.6	58.0	17.1	17.1	23.9
FacesUCR	9.4	8.5	25.6	27.7	25.7	26.9	88.4	28.4	23.5	25.8	7.4	7.4	9.6	23.1	8.3	15.6	35.5	14.3	9.5	8.8	23.1	11.1	4.3	23.1	54.7	8.2	9.8	15.8
FiftyWords	24.0	25.1	42.0	36.5	72.7	39.6	94.9	44.4	33.6	38.7	24.0	22.9	24.6	36.9	26.6	30.3	48.4	30.1	31.0	24.2	36.9	23.1	26.6	36.9	77.8	33.9	26.6	49.7
Fish	5.7	5.7	46.3	38.9	60.0	21.1	82.3	25.7	21.7	20.6	17.1	6.3	5.7	19.4	8.0	7.4	18.9	7.4	17.7	15.4	21.7	5.7	12.0	21.7	71.4	30.3	21.1	50.3
FordA	22.6	13.1	22.6	13.5	9.2	33.9	10.8	24.5	21.4	29.1	27.6	24.3	27.7	32.0	15.3	27.7	32.0	15.3	44.5	30.9	33.5	44.8	40.8	33.5	5.1	41.8	37.6	39.0
FordB	32.2	27.0	34.3	25.3	25.9	43.3	49.5	39.4	36.7	38.1	35.4	33.1	33.3	41.0	29.4	34.8	41.0	27.9	38.0	39.3	39.4	41.9	38.1	39.4	22.6	40.1	39.0	42.1
FreezerRegularTrain	8.8	33.5	10.9	42.4	16.1	19.6	45.7	17.2	19.2	20.5	10.1	10.3	9.7	19.5	33.5	9.3	31.7	40.5	10.1	9.3	19.5	2.1	8.9	19.5	40.1	3.0	7.5	8.2
FreezerSmallTrain	16.5	28.8	16.8	33.2	29.7	32.6	48.9	32.5	32.4	32.3	24.1	17.5	16.5	32.3	28.8	14.9	40.3	31.2	24.1	32.4	32.4	26.3	29.6	32.4	33.7	18.1	24.4	30.7
Fungi	16.1	16.7	54.3	14.0	3.2	16.1	78.0	19.4	15.6	21.0	16.1	16.1	16.1	17.7	16.7	30.6	27.4	9.7	16.1	17.7	17.7	2.2	3.2	17.7	59.1	11.3	16.1	19.4
GestureMidAirD1	36.9	34.6	83.1	82.3	79.2	56.9	91.5	58.5	43.8	45.4	38.5	38.5	36.9	42.3	33.1	50.0	65.4	46.9	43.1	36.1	42.3	43.1	41.5	42.3	94.6	43.9	41.5	62.3
GestureMidAirD2	**39.2**	61.5	77.7	71.5	75.4	48.5	94.6	44.6	47.7	60.0	40.0	40.0	**39.2**	50.8	63.1	63.1	63.9	76.2	**39.2**	40.0	50.8	43.1	45.4	50.8	88.5	55.4	**39.2**	60.8
GestureMidAirD3	67.7	**62.3**	87.7	86.2	91.5	76.9	93.1	70.8	63.1	67.7	63.1	63.1	67.7	67.7	**62.3**	79.2	71.5	75.4	67.7	**62.3**	65.4	70.8	65.4	65.4	90.8	73.8	66.9	75.4
GesturePebbleZ1	20.9	11.6	81.4	77.9	59.9	37.2	77.9	23.3	36.6	59.9	12.8	12.8	20.9	26.7	12.8	81.4	83.7	77.9	20.9	17.4	26.7	19.2	10.5	26.7	75.0	15.1	21.5	27.3
GesturePebbleZ2	32.9	19.0	80.4	83.5	58.2	41.8	65.8	31.0	31.6	62.7	27.2	27.2	32.9	32.9	16.5	76.0	84.8	74.0	32.9	22.1	32.9	20.9	20.9	32.9	76.6	22.8	29.1	42.4
GunPoint	2.0	3.3	8.0	7.3	8.0	8.7	51.3	10.0	7.3	7.3	8.7	1.3	2.0	8.7	6.0	1.3	8.7	6.0	9.3	8.7	8.7	3.3	4.0	8.7	21.3	10.7	11.3	9.3
GunPointAgeSpan	**0.9**	3.5	5.7	10.4	10.4	9.8	28.8	12.0	8.2	9.5	5.1	1.3	**0.9**	10.1	5.4	1.9	8.9	4.8	8.2	3.5	10.1	1.6	4.7	10.1	15.2	5.7	7.0	7.0
GunPointMaleVersusFemale	0.3	1.9	3.5	4.8	3.8	2.2	12.0	2.8	1.6	2.5	0.3	0.3	0.3	2.5	2.2	3.5	4.8	1.3	0.3	2.5	2.5	0.6	0.9	2.5	16.5	0.3	0.3	8.5
GunPointOldVersusYoung	4.8	**1.3**	4.1	4.4	10.2	4.8	30.8	4.8	4.8	6.7	6.0	4.8	4.8	10.2	4.1	4.8	12.7	2.5	16.2	3.5	4.8	4.4	3.2	4.8	8.6	**1.3**	5.4	11.1
Ham	49.5	45.7	56.2	42.9	46.7	39.0	51.4	41.9	48.6	44.8	52.4	52.4	52.4	41.0	45.7	54.3	43.8	50.5	53.3	40.0	40.0	40.0	46.7	40.0	51.4	38.1	52.4	41.9
HandOutlines	12.7	15.9	31.9	30.5	40.0	13.2	43.0	13.5	13.8	13.8	11.9	13.0	13.2	13.5	16.2	26.8	16.2	15.9	11.9	13.8	13.8	12.2	**11.1**	13.8	38.4	13.0	12.2	18.9
Haptics	60.1	68.2	73.4	72.4	75.0	61.7	78.6	62.0	58.4	58.4	60.1	58.1	60.1	56.5	68.2	71.1	68.5	75.7	62.3	58.8	63.0	61.0	59.4	63.0	74.0	59.1	59.4	68.2
Herring	42.2	46.9	53.1	51.6	48.4	43.8	46.9	54.7	45.3	46.9	40.6	40.6	45.3	48.4	40.6	56.2	51.6	50.0	46.9	46.9	48.4	42.2	46.9	48.4	48.4	48.4	50.0	51.6
HouseTwenty	7.6	10.1	14.3	24.4	13.4	57.1	42.0	14.3	21.0	31.9	7.6	7.6	7.6	33.6	10.1	16.0	58.0	39.5	7.6	5.9	33.6	20.2	11.8	33.6	37.0	11.8	7.6	26.9
InlineSkate	39.3	51.6	49.1	62.5	79.8	66.9	82.0	66.7	62.9	64.7	62.0	47.5	43.8	65.8	61.5	49.1	76.5	62.4	61.6	61.3	65.8	62.9	61.8	65.8	20.2	60.5	61.6	72.2
InsectEPGRegularTrain	7.2	25.3	7.6	48.2	21.7	33.3	39.8	25.7	36.5	30.5	20.9	13.2	9.2	32.9	30.5	5.2	49.4	53.4	12.8	17.3	32.1	24.1	23.7	32.1	9.6	14.1	12.8	28.5
InsectEPGSmallTrain	14.1	32.9	13.2	47.0	24.5	34.5	40.2	27.3	41.0	35.3	26.5	20.5	16.5	33.7	34.5	**4.8**	51.8	53.4	26.5	30.5	33.7	27.7	27.7	33.7	16.9	20.5	25.7	36.5
InsectWingbeatSound	64.8	42.4	81.9	55.8	57.4	42.7	89.0	56.9	43.7	44.4	42.0	43.8	64.5	43.6	42.3	76.2	49.0	49.9	64.5	**41.5**	43.8	44.4	51.7	43.8	85.3	61.1	52.0	56.5

(*continued on next page*)

Table 4 (*continued*).

ItalyPowerDemand	5.0	6.7	16.5	12.3	8.4	4.0	48.0	4.9	4.4	4.4	5.0	5.0	5.0	4.5	5.7	12.2	**2.7**	6.8	5.0	4.5	4.5	10.8	5.8	4.5	29.3	7.5	5.0	9.1
LargeKitchenAppliances	19.7	42.1	30.4	59.2	20.5	60.0	69.9	38.7	47.7	60.3	21.9	21.9	20.5	54.1	42.1	26.1	60.0	57.3	20.5	20.5	50.7	48.0	35.5	50.7	44.0	20.8	20.5	42.4
Lightning2	13.1	18.0	45.9	42.6	23.0	39.3	39.3	16.4	24.6	29.5	9.8	9.8	13.1	27.9	18.0	36.1	49.2	45.9	13.1	13.1	24.6	16.4	13.1	24.6	36.1	9.8	13.1	24.6
Lightning7	32.9	45.2	48.0	68.5	47.9	49.3	63.0	26.0	39.7	46.6	28.8	28.8	32.9	43.8	46.6	46.6	68.5	79.5	27.4	28.8	42.5	39.7	28.8	42.5	68.5	32.9	24.7	39.7
Mallat	5.1	6.0	19.3	9.6	53.3	8.4	86.5	7.3	7.5	8.6	5.9	5.3	5.1	8.6	9.0	11.8	14.8	8.8	6.6	8.6	8.6	15.4	7.5	8.6	35.0	11.5	5.9	16.1
Meat	6.7	53.3	50.0	53.3	13.3	6.7	61.7	6.7	6.7	6.7	6.7	6.7	6.7	6.7	61.7	31.7	28.3	61.7	6.7	6.7	6.7	26.7	6.7	6.7	53.3	6.7	6.7	10.0
MedicalImages	26.3	34.5	36.7	45.0	35.4	32.2	66.3	27.2	30.9	31.4	26.3	26.3	26.3	31.6	33.2	34.1	39.0	42.2	26.3	25.3	31.6	34.1	28.7	31.6	41.4	28.7	26.1	36.8
MelbournePedestrian	19.0	31.5	33.4	47.0	25.0	15.7	73.3	18.7	**14.7**	15.4	18.2	17.6	19.3	15.6	37.9	23.8	18.5	40.1	20.9	18.4	15.3	24.1	18.4	15.0	76.3	21.3	19.7	22.0
MiddlePhalanxOutlineAgeGroup	48.0	61.0	44.2	57.8	51.9	48.7	59.7	48.7	48.7	47.4	50.0	48.0	46.1	47.4	61.0	47.4	46.8	61.0	50.0	48.0	48.1	53.9	50.0	48.7	63.6	46.1	52.6	51.3
MiddlePhalanxOutlineCorrect	26.1	28.9	29.2	34.0	29.9	23.0	47.1	25.8	23.7	22.3	30.2	26.8	29.6	21.6	43.0	25.1	21.6	37.5	30.2	23.4	23.4	30.9	24.7	23.0	40.2	28.9	26.8	29.6
MiddlePhalanxTW	51.3	53.2	50.0	53.2	51.3	49.4	80.5	48.0	50.0	48.7	49.4	52.0	51.3	51.3	53.9	50.6	51.3	53.9	49.4	49.4	48.7	46.1	**40.9**	49.4	61.0	51.3	48.7	50.0
MixedShapesRegularTrain	3.4	**3.3**	9.9	17.6	49.0	12.5	76.8	17.5	6.6	13.2	10.7	3.4	**3.3**	10.3	4.7	4.0	29.9	5.1	15.8	9.1	10.3	10.8	8.7	10.3	19.4	14.0	13.4	23.5
MixedShapesSmallTrain	7.8	7.9	15.8	21.9	56.2	19.1	76.9	26.1	13.0	19.5	20.0	**6.3**	8.7	16.7	8.4	8.7	39.4	9.4	22.0	16.7	16.5	15.6	14.8	16.5	23.8	20.6	21.4	32.8
MoteStrain	16.7	15.2	38.0	35.8	28.0	19.4	42.2	21.3	21.2	17.3	12.4	12.9	16.7	26.3	15.2	30.5	28.4	27.0	16.5	13.4	12.1	12.3	**9.8**	12.1	44.0	13.8	21.9	25.2
NonInvasiveFetalECGThorax1	17.7	25.6	60.5	65.3	47.0	17.0	95.8	16.1	16.3	18.7	20.5	19.3	19.4	17.4	37.2	31.0	45.5	37.6	21.0	18.9	17.1	23.1	19.1	17.1	92.3	19.3	20.1	43.7
NonInvasiveFetalECGThorax2	11.2	17.5	42.2	61.2	39.6	12.2	96.3	12.4	12.1	12.5	13.4	10.7	10.7	12.0	21.5	17.2	21.8	22.2	13.5	12.9	12.0	14.7	**10.5**	12.0	90.0	15.0	13.0	36.3
OliveOil	16.7	83.3	63.3	83.3	23.3	16.7	43.3	13.3	13.3	13.3	16.7	16.7	16.7	13.3	83.3	13.3	16.7	83.3	16.7	13.3	13.3	56.7	16.7	13.3	63.3	13.3	20.0	16.7
OSULeaf	12.4	14.1	26.0	40.1	52.1	46.7	76.9	50.4	43.8	46.3	38.0	**9.5**	12.0	48.4	17.4	11.6	60.7	14.1	40.9	38.8	47.9	28.9	40.9	47.9	36.0	44.6	40.1	66.1
PhalangesOutlinesCorrect	27.3	25.5	28.2	30.4	28.9	24.1	43.9	25.1	23.8	23.1	27.2	25.4	26.1	21.0	39.6	26.0	21.0	39.5	27.2	23.9	23.9	29.3	24.9	23.9	48.0	27.5	27.2	30.8
Phoneme	73.2	**70.4**	80.2	81.7	86.2	92.0	93.1	81.3	88.3	88.1	76.2	72.7	73.1	88.8	71.7	75.5	91.6	76.2	77.2	77.3	89.1	81.0	78.1	89.1	79.0	80.1	77.0	83.4
PickupGestureWiimoteZ	38.0	40.0	56.0	76.0	66.0	52.0	92.0	40.0	34.0	52.0	34.0	38.0	32.0	44.0	36.0	54.0	90.0	46.0	34.0	34.0	44.0	36.0	34.0	44.0	76.0	32.0	34.0	38.0
PigAirwayPressure	82.7	81.7	86.1	81.2	91.3	93.3	98.6	90.9	90.4	94.7	89.9	83.2	81.2	94.2	86.1	81.2	97.1	84.1	89.4	90.4	94.2	91.3	90.9	94.2	42.8	93.3	92.3	93.8
PigArtPressure	50.0	18.3	86.5	60.1	34.1	80.8	91.8	95.2	81.2	87.0	74.0	42.8	50.0	88.0	25.5	47.6	92.3	26.4	75.5	80.3	87.5	73.1	77.9	87.5	66.8	71.6	78.4	88.5
PigCVP	76.9	53.9	95.2	87.5	66.3	82.7	97.6	89.9	88.0	89.9	81.7	76.0	76.9	92.8	55.8	81.2	95.2	55.8	84.6	84.1	91.8	84.6	88.0	91.8	53.8	81.7	84.1	92.3
PLAID	16.2	24.2	31.5	32.2	30.9	56.2	68.3	30.0	34.8	50.1	15.5	15.5	16.2	46.4	27.4	32.2	63.9	32.2	16.2	16.6	46.6	24.6	23.6	46.7	73.6	17.7	16.0	42.3
Plane	**0.0**	**0.0**	**0.0**	**0.0**	3.8	3.8	74.3	2.9	3.8	3.8	**0.0**	**0.0**	**0.0**	3.8	**0.0**	**0.0**	2.9	**0.0**	**0.0**	**0.0**	3.8	**0.0**	**0.0**	3.8	11.4	**0.0**	**0.0**	15.2
PowerCons	13.9	12.8	33.3	26.7	19.4	15.6	30.0	11.1	10.6	10.0	12.8	12.8	13.9	**6.1**	13.9	27.8	21.7	23.3	12.2	7.8	6.7	13.3	11.1	6.7	31.1	12.2	10.6	14.4
ProximalPhalanxOutlineAgeGroup	20.0	20.5	19.5	21.5	22.9	20.5	65.4	22.4	21.0	20.5	19.5	20.5	20.0	21.5	20.5	20.5	18.1	20.5	19.5	21.5	21.5	23.9	20.5	21.5	45.9	19.0	23.4	23.9
ProximalPhalanxOutlineCorrect	23.0	20.3	22.3	21.6	25.8	19.6	32.6	22.0	19.2	18.6	21.6	20.6	20.6	16.1	31.6	21.0	15.5	36.4	21.6	21.0	19.2	21.6	21.6	19.6	38.1	25.8	22.7	25.4
ProximalPhalanxTW	28.3	24.9	27.8	30.2	33.2	29.3	77.6	31.2	29.3	30.2	24.4	**22.9**	23.4	24.9	53.7	26.3	24.4	53.7	24.4	24.4	29.3	29.3	29.8	29.3	36.1	24.9	24.9	28.3
RefrigerationDevices	53.9	54.1	58.9	51.5	59.5	62.1	67.5	57.9	58.9	60.8	53.9	53.1	55.5	60.5	53.9	57.6	66.9	61.1	53.6	56.0	60.5	50.9	51.2	60.5	66.4	59.5	53.6	59.7
Rock	**8.0**	16.0	36.0	44.0	68.0	22.0	62.0	28.0	26.0	18.0	30.0	12.0	**8.0**	16.0	30.0	**8.0**	10.0	26.0	40.0	16.0	16.0	34.0	22.0	16.0	44.0	36.0	40.0	54.0
ScreenType	52.3	54.1	49.3	58.7	61.6	70.4	68.0	63.2	58.7	65.3	57.1	54.4	57.1	64.0	54.4	56.3	65.3	54.1	60.3	58.9	64.0	60.8	65.6	64.0	49.6	61.6	58.1	63.5
SemgHandGenderCh2	19.7	54.5	29.8	56.3	39.8	34.2	48.3	25.2	33.0	29.8	20.3	20.3	19.8	23.8	54.8	30.8	58.7	59.7	19.8	15.5	23.8	31.5	22.8	23.8	40.2	12.2	12.5	20.5
SemgHandMovementCh2	41.8	77.6	67.1	78.7	66.9	82.4	75.8	43.6	59.1	69.6	46.4	46.4	41.8	63.1	78.2	63.8	78.0	78.4	41.6	36.2	63.1	67.1	45.6	63.1	78.4	32.7	**32.2**	43.1
SemgHandSubjectCh2	27.3	69.6	48.2	72.2	53.3	77.8	72.4	30.7	44.7	59.1	27.3	27.3	27.3	59.6	67.6	48.7	77.8	78.0	27.3	20.0	59.6	44.9	25.1	59.6	62.7	14.9	**12.9**	26.0
ShakeGestureWiimoteZ	14.0	18.0	56.0	48.0	40.0	50.0	88.0	26.0	42.0	40.0	14.0	**12.0**	14.0	40.0	14.0	34.0	78.0	38.0	14.0	16.0	40.0	20.0	20.0	40.0	76.0	14.0	14.0	38.0
ShapeletSim	39.4	15.6	42.2	27.2	13.3	31.1	34.4	31.7	23.9	52.2	15.6	15.6	38.9	53.3	15.6	43.3	55.0	22.2	35.0	30.0	46.1	27.2	20.0	48.3	20.6	39.4	41.1	47.2
ShapesAll	15.0	13.0	57.0	63.3	63.3	27.3	95.5	32.0	22.7	24.2	19.7	14.0	15.0	24.8	12.8	18.5	47.0	13.8	23.2	19.8	24.8	16.2	18.3	24.8	68.5	24.7	20.8	41.7
SmallKitchenAppliances	36.0	63.7	38.9	61.3	50.7	64.3	47.7	56.8	60.8	69.3	35.2	36.0	36.0	63.5	70.1	36.3	66.4	67.7	35.7	32.8	65.9	60.3	38.1	65.6	68.0	35.5	31.5	36.5
Strawberry	4.9	27.0	7.3	30.0	18.1	5.4	46.8	5.7	5.7	6.2	6.0	**4.3**	4.6	4.6	35.7	**4.3**	4.6	36.8	6.0	5.4	5.4	10.3	6.5	5.4	22.2	5.1	5.7	10.3
SwedishLeaf	9.9	10.6	21.9	23.0	24.8	25.1	86.9	22.9	12.3	19.4	20.2	10.1	9.9	21.1	16.5	11.8	44.2	16.5	20.8	15.4	21.1	14.1	12.0	21.1	38.2	19.5	17.4	45.4
Symbols	4.7	3.2	41.3	76.4	16.7	10.3	75.7	23.7	8.4	11.8	5.0	5.3	4.7	18.7	3.8	2.5	21.5	3.8	5.0	6.2	10.1	5.9	6.1	10.1	43.7	3.1	5.0	17.7
SyntheticControl	0.7	6.0	69.7	63.7	36.7	48.0	83.7	6.3	5.0	25.7	1.3	1.3	0.7	12.0	6.0	50.0	65.3	48.3	0.7	1.7	12.0	18.7	3.3	11.3	38.7	3.3	0.7	3.3
ToeSegmentation1	19.3	23.7	38.6	41.7	20.6	39.0	39.0	27.6	30.7	35.5	23.7	23.7	19.3	39.9	23.7	19.3	41.2	16.7	22.8	25.0	32.0	20.2	21.5	32.0	34.2	22.8	26.3	29.8
ToeSegmentation2	25.4	11.5	48.5	21.5	13.8	29.2	41.5	20.0	18.5	16.9	13.8	17.7	25.4	20.8	11.5	28.5	66.2	13.1	16.1	9.2	19.2	10.8	9.2	19.2	26.2	20.0	16.1	21.5
Trace	**0.0**	4.0	23.0	43.0	20.0	21.0	75.0	9.0	14.0	28.0	**0.0**	**0.0**	**0.0**	23.0	4.0	1.0	41.0	1.0	**0.0**	1.0	24.0	10.0	17.0	24.0	**0.0**	**0.0**	**0.0**	26.0
TwoLeadECG	2.2	6.9	3.2	15.4	15.1	24.1	19.6	28.5	23.2	22.1	9.6	2.5	2.2	20.2	18.4	1.0	15.3	8.5	9.6	13.2	25.3	5.6	6.1	25.2	27.3	16.2	9.6	32.1
TwoPatterns	**0.0**	0.1	3.0	34.9	52.0	54.8	74.8	40.5	12.1	31.8	**0.0**	**0.0**	**0.0**	9.3	0.1	0.3	61.3	47.2	**0.0**	0.1	9.3	0.5	**0.0**	9.3	74.5	**0.0**	**0.0**	**0.0**
UMD	**0.7**	10.4	1.4	17.4	29.9	36.8	29.2	20.8	13.9	25.7	**0.7**	**0.7**	**0.7**	20.8	11.8	4.2	22.2	11.8	**0.7**	2.8	23.6	9.0	5.6	23.6	30.6	1.4	**0.7**	17.4
UWaveGestureLibraryAll	6.7	4.3	39.7	56.1	63.2	12.5	87.2	11.8	6.0	6.4	4.0	3.7	6.5	5.2	4.3	16.9	58.5	30.2	10.8	**3.4**	5.2	4.7	3.9	5.2	79.1	11.2	4.8	10.5
UWaveGestureLibraryX	21.4	20.0	62.3	73.8	59.8	46.2	87.5	39.1	23.8	25.6	23.6	20.2	22.1	26.1	20.5	33.2	39.3	28.1	27.2	22.7	26.1	25.3	24.2	26.1	64.7	25.7	24.1	33.8
UWaveGestureLibraryY	28.7	28.3	67.3	74.6	71.6	43.8	87.0	43.4	29.0	34.6	30.9	25.6	28.4	33.8	28.6	41.0	58.3	39.2	36.6	30.1	33.8	33.4	31.1	33.8	63.8	37.2	32.5	41.7
UWaveGestureLibraryZ	28.7	25.8	60.6	69.4	**0.0**	**0.0**	85.1	40.7	**0.0**	**0.0**	32.1	28.3	30.4	35.1	27.5	42.0	44.8	34.4	34.2	32.2	35.0	32.7	31.5	35.0	**0.0**	35.6	32.7	43.4
Wafer	2.0	0.5	1.9	0.4	1.1	0.6	9.1	0.7	0.6	0.5	0.5	0.5	2.0	0.5	0.5	2.0	0.9	0.8	2.0	0.4	0.5	1.5	0.9	0.5	8.2	2.4	1.2	1.6
Wine	42.6	50.0	57.4	50.0	51.9	46.3	48.1	38.9	38.9	38.9	42.6	42.6	42.6	38.9	50.0	50.0	50.0	50.0	42.6	38.9	38.9	50.0	35.2	38.9	53.7	38.9	42.6	50.0
WordSynonyms	26.8	27.9	46.7	40.1	70.4	40.8	87.5	47.8	35.7	41.2	25.6	**24.0**	27.0	38.4	27.4	30.9	50.5	31.8	35.1	26.2	38.2	31.3	30.6	38.2	76.5	32.8	31.8	53.6
Worms	41.6	37.7	63.6	57.1	37.7	50.6	57.1	44.2	46.8	49.4	49.4	40.3	41.6	54.5	36.4	33.8	57.1	42.9	41.6	46.8	54.5	40.3	44.2	54.5	28.6	50.6	41.6	61.0
WormsTwoClass	37.7	33.8	49.4	44.2	27.3	33.8	37.7	31.2	39.0	40.3	41.6	41.6	35.1	39.0	26.0	26.0	39.0	23.4	37.7	41.6	39.0	29.9	33.8	39.0	20.8	41.6	36.4	44.2
Yoga	14.4	12.3	31.8	20.1	33.4	16.5	48.6	18.1	16.4	17.7	15.1	14.1	14.4	17.3	15.0	18.0	25.1	15.0	16.4	15.6	17.0	13.5	14.1	17.0	36.8	17.3	16.1	23.3
Number of wins	9	7	3	1	1	4	0	1	4	2	8	13	11	2	4	5	1	2	8	5	1	2	9	1	2	6	9	1
Average rank	15.2	19.5	34.5	38.7	35.0	29.7	51.4	27.0	22.5	28.1	16.7	12.7	15.4	26.3	22.2	25.9	39.1	29.6	21.5	15.7	26.1	22.3	16.3	26.2	42.8	21.6	20.4	36.1
Mean error rate	23.2	26.7	39.7	43.4	38.2	33.9	63.2	30.5	28.7	32.3	24.6	22.2	23.3	30.8	29.0	30.7	43.2	35.4	26.6	24.3	30.6	27.9	24.7	30.6	47.2	26.4	25.8	35.3
Median error rate	20.0	20.5	38.6	42.0	34.8	33.8	66.3	27.6	26.0	29.5	21.9	20.0	20.0	26.7	26.0	29.5	41.9	32.2	23.2	20.5	26.7	25.1	21.6	26.7	44.0	22.8	24.1	35.0
Standard deviation of the error rate	18.3	21.1	23.1	23.1	22.3	21.5	22.8	19.5	19.9	21.2	18.8	17.8	18.2	20.5	21.8	21.2	24.0	23.6	19.0	18.8	20.4	20.0	19.3	20.5	23.6	19.5	19.1	20.3
Max error rate	82.7	83.3	95.2	87.5	91.5	93.3	98.6	95.2	90.4	94.7	89.9	83.2	81.	94.2	86.1	81.4	97.1	85.7	89.4	90.4	94.2	91.3	90.9	94.2	94.6	93.3	92.3	93.8

Table 5

Error rates (in %) of all considered distance measures using the 1NN classifier. The best-performing classifier for each dataset is highlighted in bold. The final six rows show the number of wins per distance, average ranks, mean and median error rates, the standard deviation of the error rates, and the maximum error rate.

Dataset	Inf	IP	ISD	LCSS	LLR_DLS	LLR_LK	LNP	MAN	MIN	MPdist	MSM	NCD	NP	P	PACF	PC2	PDC	PIC	SAX	SDTW	$\text{shapeDTW}_{\text{HOG1D}}$	$\text{shapeDTW}_{\text{RawSub}}$	STS	TAM	$\text{TD}^{\text{C}}_{\text{DTW}}$	$\text{TD}^{\text{H}}_{\text{DTW}}$	$\text{TD}^{\text{S}}_{\text{DTW}}$	TQ
ACSF1	67.0	35.0	37.0	83.0	29.0	29.0	52.0	**16.0**	56.0	29.0	17.0	58.0	36.0	36.0	34.0	46.0	**16.0**	44.0	95.0	29.0	27.0	26.0	46.0	67.0	34.0	38.0	34.0	79.0
Adiac	38.4	44.0	76.2	97.2	62.9	62.9	68.5	40.2	38.1	35.8	38.6	92.8	50.9	50.9	45.5	38.9	90.3	83.6	97.4	38.9	**26.9**	33.5	41.9	39.6	38.6	37.6	38.4	76.2
AllGestureWiimoteX	63.3	59.1	67.7	35.0	67.6	67.6	76.4	45.1	51.4	35.3	44.3	84.9	54.6	54.6	74.6	48.4	81.1	82.9	69.3	45.7	30.1	25.7	69.9	70.9	28.4	28.4	26.3	64.0
AllGestureWiimoteY	56.7	57.3	69.3	30.4	68.7	68.7	79.3	41.6	46.3	29.3	41.7	79.7	49.4	49.4	70.7	43.1	85.7	83.4	70.7	41.7	26.5	**23.9**	70.4	72.3	25.7	25.9	26.1	59.1
AllGestureWiimoteZ	69.1	55.0	68.9	42.9	68.7	68.7	78.9	51.7	57.9	44.3	52.1	84.0	49.7	49.7	66.0	54.6	84.0	78.6	70.4	48.4	**30.0**	35.3	78.0	70.7	32.3	35.7	34.1	70.0
ArrowHead	17.7	21.1	41.1	57.7	36.6	36.6	35.4	20.6	18.9	20.0	27.4	65.7	23.4	23.4	41.1	20.0	51.4	57.7	48.6	24.0	17.7	19.4	40.6	26.3	21.7	26.3	18.3	43.4
Beef	30.0	53.3	63.3	66.7	60.0	60.0	33.3	36.7	33.3	30.0	46.7	76.7	56.7	56.7	46.7	33.3	66.7	53.3	70.0	33.3	**26.7**	40.0	30.0	50.0	36.7	33.3	33.3	56.7
BeetleFly	35.0	10.0	50.0	35.0	50.0	50.0	25.0	35.0	25.0	35.0	30.0	55.0	25.0	25.0	35.0	25.0	65.0	50.0	20.0	40.0	20.0	30.0	35.0	20.0	25.0	30.0	35.0	15.0
BirdChicken	35.0	15.0	5.0	25.0	15.0	15.0	5.0	40.0	45.0	20.0	25.0	40.0	25.0	25.0	45.0	45.0	25.0	40.0	35.0	20.0	5.0	25.0	45.0	30.0	25.0	25.0	25.0	45.0
BME	24.0	31.3	37.3	11.3	36.7	36.7	48.0	14.7	16.0	12.7	17.3	51.3	32.0	32.0	45.3	16.7	62.0	43.3	30.7	15.3	**0.0**	3.3	25.3	31.3	3.3	2.0	4.0	18.7
Car	28.3	31.7	41.7	56.7	56.7	56.7	56.7	28.3	26.7	21.7	**10.0**	78.3	45.0	45.0	66.7	26.7	53.3	70.0	60.0	28.3	13.3	11.7	33.3	20.0	26.7	26.7	30.0	35.0
CBF	57.8	38.0	43.3	4.0	41.8	41.8	51.6	11.1	19.9	4.4	1.2	40.7	31.2	31.2	58.8	14.8	60.6	49.9	20.9	5.9	8.0	1.6	66.0	0.8	0.3	0.3	0.3	32.0
Chinatown	8.2	11.1	19.5	72.5	14.3	14.3	21.0	3.2	5.5	3.8	3.5	38.8	15.7	15.7	15.7	4.7	40.2	26.8	44.3	5.5	10.1	**3.2**	9.9	9.9	5.2	4.1	5.2	47.8
ChlorineConcentration	34.2	42.3	37.8	61.5	35.1	35.1	28.6	37.3	33.2	34.7	37.2	52.6	38.7	38.7	29.4	35.0	37.9	56.5	58.6	35.1	35.5	35.5	27.7	40.7	35.1	35.9	35.0	45.3
CinCECGTorso	20.3	36.4	11.2	7.1	26.7	26.7	8.6	5.6	12.9	28.3	23.8	48.1	18.7	18.7	15.4	10.3	**1.4**	37.8	31.0	15.4	20.9	24.8	21.4	19.6	20.0	30.4	21.7	30.7
Coffee	**0.0**	**0.0**	17.9	46.4	14.3	14.3	21.4	3.6	**0.0**	7.1	7.1	42.9	7.1	7.1	**0.0**	**0.0**	42.9	14.3	46.4	**0.0**	3.6	3.6	7.1	17.9	**0.0**	**0.0**	**0.0**	32.1
DiatomSizeReduction	3.6	6.5	22.2	69.9	44.4	44.4	25.5	7.2	5.6	9.2	4.6	69.9	6.5	6.5	11.8	6.5	34.6	39.2	73.5	6.2	6.9	3.9	11.1	9.2	**3.3**	**3.3**	**3.3**	12.1
DistalPhalanxOutlineAgeGroup	36.7	37.4	35.3	38.1	31.7	31.7	41.7	32.4	36.7	23.8	29.5	57.6	36.0	36.0	44.6	37.4	40.3	38.1	57.6	33.1	29.5	33.8	39.6	32.4	33.1	28.8	30.2	36.0
DistalPhalanxOutlineCorrect	35.1	29.3	28.6	39.9	33.7	33.7	31.2	25.7	30.1	24.3	26.1	42.4	31.5	31.5	29.3	28.3	47.1	39.1	46.7	26.4	**24.3**	29.0	25.4	30.8	26.4	26.8	26.4	35.9
DistalPhalanxTW	41.7	44.6	47.5	49.6	44.6	44.6	50.4	37.4	39.6	**27.8**	36.7	73.4	45.3	45.3	46.0	36.7	49.6	43.9	74.1	37.4	41.0	36.7	46.0	45.3	39.6	36.7	40.3	43.2
DodgerLoopDay	65.0	71.2	81.2	50.0	80.0	80.0	83.8	42.5	48.8	51.2	51.2	87.5	51.2	51.2	78.8	45.0	78.8	85.0	58.8	40.0	62.5	40.0	83.8	48.8	42.5	43.8	32.5	58.8
DodgerLoopGame	15.2	42.0	49.3	11.6	47.1	47.1	47.1	11.6	10.9	22.5	16.7	52.2	16.7	16.7	39.9	11.6	52.9	40.6	18.8	9.4	18.1	8.0	49.3	15.2	13.0	10.1	9.4	27.5
DodgerLoopWeekend	8.7	23.2	46.4	1.4	37.0	37.0	44.9	1.4	2.2	10.1	2.2	26.1	8.0	8.0	31.2	1.4	58.7	37.0	3.6	2.2	7.2	2.2	27.5	4.3	2.2	5.1	2.9	2.2
Earthquakes	38.8	33.1	37.4	31.6	36.0	36.0	28.8	28.8	31.7	26.7	34.5	31.7	38.1	38.1	33.1	28.8	30.9	34.5	38.8	30.9	37.4	30.2	69.8	33.8	30.2	28.8	28.1	33.1
ECG200	13.0	20.0	15.0	12.0	16.0	16.0	19.0	11.0	11.0	14.0	16.0	37.0	22.0	22.0	20.0	12.0	38.0	24.0	30.0	**8.0**	10.0	14.0	15.0	14.0	16.0	20.0	9.0	22.0
ECG5000	7.9	8.8	11.6	**6.6**	11.2	11.2	10.2	7.4	7.4	6.9	6.9	24.0	8.7	8.7	11.9	7.5	35.4	30.7	11.8	7.8	7.1	7.0	7.7	8.2	7.6	7.6	7.6	10.6
ECGFiveDays	28.9	3.0	25.6	5.7	17.4	17.4	16.7	21.5	20.7	3.5	19.9	50.1	**0.1**	**0.1**	22.9	20.3	38.3	31.6	45.6	21.2	5.7	7.9	38.3	22.5	17.0	21.1	19.3	22.6
FaceAll	33.8	41.7	55.6	24.4	54.0	54.0	51.4	27.9	23.4	25.9	23.2	91.0	33.2	33.2	31.8	28.6	87.5	60.8	45.6	28.9	23.8	21.7	28.5	20.9	19.5	20.3	19.6	31.3
FaceFour	45.5	40.9	51.1	18.2	52.3	52.3	43.2	15.9	27.3	5.7	12.5	65.9	34.1	34.1	55.7	21.6	53.4	69.3	21.6	36.4	9.1	10.2	47.7	14.8	17.1	14.8	17.1	12.5
FacesUCR	40.9	41.5	51.5	10.0	50.9	50.9	51.4	19.9	26.8	11.3	5.8	88.8	22.5	22.5	40.5	23.1	84.6	64.8	37.9	29.8	8.1	**3.4**	35.5	6.6	8.9	8.3	9.7	25.5
FiftyWords	47.5	66.6	73.4	31.6	80.7	80.7	71.6	33.2	39.8	30.5	24.6	95.4	60.0	60.0	67.5	36.9	90.1	74.5	42.0	42.0	24.2	**20.2**	48.4	24.6	30.6	29.0	28.6	42.6
Fish	20.6	28.6	66.3	85.1	70.3	70.3	58.9	20.6	20.0	9.7	7.4	84.0	36.6	36.6	45.1	21.7	73.1	75.4	81.7	21.1	**5.1**	**5.1**	18.9	13.7	20.0	14.3	17.7	60.0
FordA	36.4	7.2	**3.6**	27.6	11.9	11.9	4.6	31.6	33.7	27.7	43.9	10.6	17.6	17.6	13.3	33.5	32.4	24.4	36.6	26.1	23.2	27.7	32.0	39.6	24.8	41.7	25.8	35.0
FordB	43.2	**21.0**	21.9	35.4	26.9	26.9	29.1	39.9	40.5	34.8	38.3	48.9	33.1	33.1	31.9	39.4	41.5	35.8	46.7	37.8	31.7	34.4	41.0	39.9	36.9	39.5	38.8	45.4
FreezerRegularTrain	27.5	12.0	14.6	39.9	15.7	15.7	27.9	8.9	22.3	3.3	7.2	48.2	15.5	15.5	20.2	19.5	48.8	34.8	34.2	18.2	**0.1**	5.0	31.7	9.5	17.0	3.2	10.1	31.8

(*continued on next page*)

Table 5 (*continued*).

FreezerSmallTrain	30.9	19.8	23.6	33.3	25.1	25.1	39.9	29.7	32.6	**2.5**	29.2	50.7	31.6	31.6	28.4	32.4	47.0	32.8	32.0	32.3	3.1	24.9	40.3	21.5	28.5	18.9	26.4	31.8
Fungi	26.3	18.3	56.5	16.7	60.8	60.8	60.8	12.9	19.4	88.2	5.9	76.3	3.8	3.8	31.7	17.7	93.5	55.9	32.8	20.4	2.1	**0.5**	27.4	21.0	16.1	16.1	16.1	16.7
GestureMidAirD1	65.4	63.1	83.8	43.9	81.5	81.5	75.4	46.2	46.9	40.0	49.2	90.8	61.5	61.5	67.7	42.3	77.7	86.9	74.6	50.0	**30.8**	36.1	65.4	43.8	38.5	38.5	36.1	53.1
GestureMidAirD2	69.2	58.5	80.0	72.3	66.9	66.9	59.2	48.5	54.6	40.8	47.7	93.1	59.2	59.2	79.2	50.8	81.5	85.4	77.7	46.1	48.5	40.0	63.8	52.3	43.1	**39.2**	**39.2**	75.4
GestureMidAirD3	78.5	85.4	89.2	63.9	86.2	86.2	87.7	63.1	70.8	66.9	71.5	91.5	79.2	79.2	92.3	65.4	82.3	95.4	77.7	70.0	65.4	67.7	71.5	74.6	64.6	65.4	66.9	70.8
GesturePebbleZ1	79.1	55.8	72.7	12.8	68.0	68.0	80.2	13.4	53.5	9.3	25.0	77.9	25.6	25.6	79.7	26.7	62.8	84.9	30.8	17.4	40.7	**5.8**	83.7	18.6	19.8	11.1	19.2	25.6
GesturePebbleZ2	84.2	58.9	78.5	16.5	70.9	70.9	83.5	25.3	52.5	15.8	35.4	63.3	36.1	36.1	84.2	32.9	70.9	83.5	45.6	28.5	43.7	**14.6**	84.8	29.1	24.7	20.2	26.6	35.4
GunPoint	14.7	8.7	22.7	26.7	10.0	10.0	20.0	4.7	12.0	4.7	3.3	52.0	10.7	10.7	4.7	8.7	36.0	32.7	33.3	8.0	**0.7**	1.3	8.7	10.0	6.7	9.3	7.3	19.3
GunPointAgeSpan	13.6	7.9	12.0	17.4	7.6	7.6	12.3	7.6	11.1	3.2	2.2	27.5	10.1	10.1	14.2	10.1	34.8	34.2	15.5	10.8	**0.9**	2.2	8.9	4.7	7.6	5.7	6.3	15.2
GunPointMaleVersusFemale	4.7	1.3	14.6	13.6	8.9	8.9	15.2	1.9	3.2	2.2	**0.3**	11.4	1.6	1.6	10.1	2.5	33.2	13.0	11.1	3.5	0.9	0.3	4.7	2.5	0.3	0.3	0.3	5.1
GunPointOldVersusYoung	13.3	5.7	6.7	26.7	7.0	7.0	6.0	2.9	7.9	6.0	4.4	30.2	7.3	7.3	17.1	4.8	14.9	37.1	25.7	5.7	**1.3**	4.8	12.7	11.4	5.4	4.1	4.4	16.5
Ham	46.7	48.6	51.4	47.6	53.3	53.3	55.2	48.6	44.8	39.0	41.9	45.7	40.0	40.0	54.3	40.0	55.2	45.7	39.0	40.0	45.7	45.7	43.8	42.9	43.8	39.0	46.7	**33.3**
HandOutlines	13.2	17.6	37.8	47.0	37.3	37.3	42.2	14.1	14.6	19.7	14.3	42.4	18.1	18.1	37.6	13.8	39.2	43.2	39.2	14.1	21.4	12.7	16.2	22.4	13.5	13.5	12.2	13.5
Haptics	61.0	61.7	74.0	69.2	65.9	65.9	73.7	64.0	61.0	63.3	58.8	78.2	65.3	65.3	68.8	63.0	73.1	73.4	68.5	61.0	**56.2**	57.5	68.5	64.0	59.4	59.1	59.1	62.3
Herring	50.0	46.9	50.0	40.6	46.9	46.9	57.8	40.6	48.4	40.6	45.3	50.0	**37.5**	**37.5**	40.6	48.4	51.6	42.2	43.8	43.8	50.0	**37.5**	51.6	53.1	46.9	39.1	46.9	39.1
HouseTwenty	54.6	21.0	26.1	10.9	21.8	21.8	17.6	15.1	48.7	27.7	8.4	35.3	22.7	22.7	21.0	33.6	31.9	42.0	25.2	18.5	**3.4**	8.4	58.0	5.0	7.6	7.6	10.1	39.5
InlineSkate	71.3	66.2	**18.0**	77.8	64.7	64.7	22.7	64.7	67.5	67.1	57.3	82.9	69.8	69.8	52.5	65.8	53.8	66.0	78.7	68.5	62.9	58.7	76.5	64.5	61.6	60.0	61.6	75.3
InsectEPGRegularTrain	45.8	16.1	10.0	30.5	6.0	6.0	15.3	28.5	34.1	5.2	13.7	41.4	31.7	31.7	7.2	32.1	21.7	14.9	39.4	26.9	**4.4**	12.8	49.4	13.3	12.8	11.2	12.8	35.3
InsectEPGSmallTrain	45.4	16.9	15.3	32.1	13.3	13.3	16.1	29.7	34.9	5.2	32.1	35.7	29.7	29.7	11.2	33.7	30.5	20.9	36.9	38.5	8.0	27.3	51.8	27.3	26.5	26.5	26.5	43.0
InsectWingbeatSound	46.1	55.9	83.9	42.3	80.2	80.2	79.5	43.1	43.6	49.6	51.7	89.2	47.6	47.6	73.9	43.8	86.4	84.7	47.7	54.5	58.4	53.3	49.0	63.1	52.9	59.2	50.5	44.7
ItalyPowerDemand	5.2	15.9	26.2	20.8	19.0	19.0	25.0	3.8	4.7	3.7	5.4	45.5	13.0	13.0	10.9	4.5	34.9	39.1	37.4	5.2	10.3	3.7	2.7	10.1	5.0	5.0	5.0	21.8
LargeKitchenAppliances	61.9	33.6	43.7	41.1	43.7	43.7	43.7	44.3	56.3	35.7	35.2	65.9	26.1	26.1	50.7	50.7	45.9	60.0	60.8	38.9	**16.0**	18.4	60.0	57.9	20.5	19.7	20.5	57.6
Lightning2	31.1	24.6	31.1	18.0	37.7	37.7	44.3	18.0	34.4	23.0	21.3	37.7	27.9	27.9	36.1	24.6	49.2	47.5	27.9	16.4	11.5	13.1	49.2	26.2	**8.2**	**8.2**	9.8	39.3
Lightning7	54.8	67.1	61.6	42.5	53.4	53.4	63.0	28.8	53.4	34.2	24.7	67.1	41.1	41.1	63.0	42.5	79.5	78.1	57.5	27.4	23.3	**17.8**	68.5	49.3	27.4	27.4	23.3	47.9
Mallat	12.1	15.2	31.4	45.9	15.9	15.9	25.9	7.5	9.1	11.2	10.3	87.2	13.3	13.3	28.7	8.6	40.7	38.7	55.4	8.0	6.2	6.4	14.8	11.7	4.9	6.6	**4.7**	30.7
Meat	10.0	**5.0**	58.3	66.7	53.3	53.3	51.7	6.7	6.7	13.3	8.3	60.0	**5.0**	**5.0**	28.3	6.7	53.3	31.7	70.0	6.7	10.0	6.7	28.3	11.7	6.7	6.7	6.7	18.3
MedicalImages	33.4	43.9	39.7	33.4	45.1	45.1	39.7	29.3	32.5	32.6	**24.7**	65.3	43.7	43.7	38.8	31.6	56.7	46.2	53.4	30.8	26.4	25.4	38.9	29.1	25.3	25.3	25.3	48.9
MelbournePedestrian	19.0	32.2	40.7	76.9	33.7	33.7	36.0	15.1	16.4	15.2	18.3	72.4	26.9	26.9	26.3	15.3	79.5	63.1	85.4	17.6	20.9	20.9	18.5	28.8	16.0	19.0	15.7	51.8
MiddlePhalanxOutlineAgeGroup	41.6	55.2	51.3	42.9	53.2	53.2	46.1	48.1	48.1	**25.2**	49.4	61.0	55.2	55.2	51.3	48.1	59.7	53.2	67.5	51.3	44.8	49.4	46.8	55.8	48.7	50.6	47.4	46.8
MiddlePhalanxOutlineCorrect	26.8	33.0	39.2	43.0	38.8	38.8	38.8	24.7	25.1	30.2	27.1	45.4	30.6	30.6	32.0	23.4	45.4	41.2	48.8	23.0	28.5	23.7	**21.6**	26.8	25.8	25.8	23.4	44.3
MiddlePhalanxTW	51.9	53.9	60.4	72.7	57.1	57.1	55.8	44.8	51.3	42.1	50.6	80.5	50.0	50.0	50.6	48.7	53.2	59.1	74.0	48.7	48.7	47.4	51.3	51.9	47.4	50.6	48.0	53.9
MixedShapesRegularTrain	13.8	17.6	17.0	17.5	30.6	30.6	27.8	10.8	10.6	8.6	13.7	75.4	27.7	27.7	30.8	10.3	39.5	35.7	19.2	16.4	4.8	12.9	29.9	19.2	11.1	11.7	12.7	19.3
MixedShapesSmallTrain	20.3	26.6	22.9	17.9	36.8	36.8	30.4	16.9	16.9	13.4	18.8	76.1	30.6	30.6	39.1	16.5	45.0	43.1	23.3	27.1	8.6	20.0	39.4	23.4	19.6	18.0	19.0	22.2
MoteStrain	26.4	45.3	43.3	13.5	43.1	43.1	44.0	13.4	18.4	18.3	12.4	41.7	42.1	42.1	42.6	12.1	17.5	46.2	28.5	23.2	11.0	10.1	28.4	11.6	16.5	15.7	16.1	17.2
NonInvasiveFetalECGThorax1	19.4	24.1	39.3	85.8	40.0	40.0	62.3	19.1	16.9	17.9	24.4	95.2	34.8	34.8	30.6	17.1	90.1	60.1	90.5	16.8	21.9	22.3	45.5	27.1	**15.5**	17.2	16.3	42.0
NonInvasiveFetalECGThorax2	13.5	18.4	39.7	74.7	49.3	49.3	65.0	12.5	12.2	12.5	16.0	96.0	27.8	27.8	21.5	12.0	86.9	57.0	86.6	12.1	14.0	11.0	21.8	23.1	11.6	12.6	12.0	36.0
OliveOil	16.7	16.7	63.3	83.3	50.0	50.0	63.3	16.7	13.3	**10.0**	16.7	50.0	16.7	16.7	20.0	13.3	50.0	50.0	73.3	13.3	**10.0**	13.3	16.7	16.7	13.3	16.7	13.3	66.7
OSULeaf	47.1	35.5	32.2	37.2	34.7	34.7	41.7	45.5	48.3	36.8	19.8	74.0	47.5	47.5	47.5	47.9	56.6	66.1	50.0	52.5	13.2	28.9	60.7	24.4	38.0	43.8	40.1	50.4
PhalangesOutlinesCorrect	27.2	33.0	36.0	36.0	34.7	34.7	32.2	24.8	25.1	28.2	25.8	43.7	31.5	31.5	25.9	23.9	45.5	39.7	44.6	23.7	26.1	23.5	**21.0**	31.0	24.9	27.0	24.2	41.0
Phoneme	92.9	78.8	77.0	78.2	77.5	77.5	78.1	89.4	91.6	85.3	72.4	92.8	87.4	87.4	82.6	89.1	78.5	85.8	96.1	79.4	73.6	76.1	91.6	71.8	75.5	77.7	75.5	94.9
PickupGestureWiimoteZ	62.0	54.0	66.0	30.0	70.0	70.0	90.0	38.0	52.0	32.0	48.0	74.0	36.0	36.0	80.0	44.0	76.0	80.0	60.0	48.0	32.0	**28.0**	90.0	50.0	30.0	32.0	34.0	58.0
PigAirwayPressure	94.2	63.0	40.4	91.3	58.7	58.7	43.3	94.7	94.7	**13.9**	83.2	98.1	81.2	81.2	82.2	94.2	65.9	78.8	94.7	91.3	73.1	86.5	97.1	79.3	89.9	90.4	89.9	92.3
PigArtPressure	88.5	9.1	19.7	74.0	23.1	23.1	15.4	87.0	88.0	**0.0**	32.7	90.4	36.1	36.1	41.8	87.5	40.9	61.1	88.9	96.2	6.2	49.0	92.3	49.5	75.5	72.6	75.5	88.0
PigCVP	92.8	35.1	33.7	78.4	41.3	41.3	31.7	90.9	91.8	**10.6**	46.2	96.2	35.1	35.1	27.9	91.8	76.4	58.2	90.9	90.4	34.6	58.2	95.2	54.8	85.1	81.7	83.2	92.8
PLAID	75.6	40.4	46.7	42.6	41.7	41.7	56.2	41.3	51.8	44.9	21.6	67.4	35.9	35.9	36.1	46.6	**7.8**	45.3	66.7	32.2	16.6	24.0	63.7	31.3	16.2	14.9	16.6	52.5
Plane	3.8	1.9	4.8	19.1	1.0	1.0	5.7	3.8	3.8	1.0	**0.0**	73.3	3.8	3.8	2.9	3.8	50.5	1.9	19.0	3.8	**0.0**	**0.0**	2.9	**0.0**	**0.0**	**0.0**	**0.0**	4.8
PowerCons	17.8	26.1	17.8	13.9	18.3	18.3	35.6	6.1	12.2	11.1	13.9	27.2	22.2	22.2	29.4	6.7	21.7	28.9	12.8	15.0	12.2	11.1	21.7	12.8	11.7	11.7	10.6	17.2
ProximalPhalanxOutlineAgeGroup	21.0	24.9	22.4	57.1	22.4	22.4	22.4	21.5	19.5	20.5	20.5	53.2	22.4	22.4	21.5	21.5	30.2	22.0	54.1	21.5	21.0	23.4	**18.0**	22.0	21.9	20.5	21.5	32.7
ProximalPhalanxOutlineCorrect	18.9	25.8	25.4	31.6	21.6	21.6	19.2	21.6	17.2	**14.8**	19.6	30.2	26.8	26.8	15.5	19.2	35.7	30.2	41.9	19.2	20.6	19.2	15.5	27.8	21.6	22.7	21.6	38.8
ProximalPhalanxTW	32.2	30.7	32.7	99.0	25.9	25.9	36.6	30.7	30.7	28.5	25.9	65.4	31.7	31.7	25.4	29.3	34.6	43.9	71.7	29.8	23.4	26.3	24.4	28.3	24.4	24.4	24.4	39.5
RefrigerationDevices	63.5	53.9	56.5	53.9	60.0	60.0	58.4	54.1	65.9	50.1	50.9	67.5	58.4	58.4	61.3	60.5	**48.3**	58.9	69.1	57.1	50.7	54.9	66.9	52.5	52.8	53.6	50.9	59.5
Rock	16.0	34.0	44.0	30.0	40.0	40.0	34.0	20.0	16.0	22.0	22.0	64.0	44.0	44.0	32.0	16.0	44.0	40.0	18.0	24.0	20.0	24.0	10.0	22.0	24.0	40.0	40.0	44.0
ScreenType	66.9	54.4	49.6	64.3	**46.7**	**46.7**	60.8	64.0	67.5	66.9	55.5	63.7	64.8	64.8	62.1	64.0	52.8	60.3	60.5	63.2	52.5	61.1	65.3	64.0	60.3	61.3	61.6	60.3
SemgHandGenderCh2	30.5	41.2	35.2	53.2	35.2	35.2	42.5	43.0	23.2	23.8	29.2	48.0	26.0	26.0	44.0	23.8	49.0	45.8	51.5	24.5	**11.3**	29.3	58.7	13.8	16.5	13.5	16.5	20.5
SemgHandMovementCh2	60.7	69.1	69.8	77.1	73.6	73.6	80.7	69.8	55.8	66.0	55.6	75.8	49.8	49.8	71.1	63.1	83.3	78.4	69.6	47.8	38.7	68.7	78.0	46.9	36.7	33.3	37.3	54.2
SemgHandSubjectCh2	46.4	55.8	55.1	67.6	54.2	54.2	74.2	67.3	49.1	59.6	42.0	72.7	44.0	44.0	66.0	59.6	81.8	69.1	69.3	30.4	17.1	56.9	77.8	25.8	23.1	14.9	21.6	32.7
ShakeGestureWiimoteZ	62.0	36.0	58.0	18.0	54.0	54.0	62.0	30.0	42.0	20.0	20.0	76.0	32.0	32.0	46.0	40.0	68.0	78.0	62.0	28.0	18.0	**12.0**	78.0	30.0	14.0	**12.0**	14.0	46.0
ShapeletSim	51.7	8.3	9.4	15.6	7.8	7.8	40.6	52.2	50.0	25.0	36.1	35.0	27.8	27.8	18.3	46.1	**2.8**	15.0	46.1	39.4	2.8	32.8	55.0	4.4	30.6	34.4	33.9	38.3
ShapesAll	28.2	32.3	60.0	47.5	53.3	53.3	66.8	24.0	24.7	21.2	14.3	92.8	36.5	36.5	62.0	24.8	68.2	78.7	46.5	31.3	**11.2**	16.3	47.0	21.8	18.8	21.0	18.7	33.8
SmallKitchenAppliances	61.1	56.3	51.2	70.1	52.8	52.8	61.3	60.3	67.5	48.5	36.8	40.3	60.0	60.0	58.1	65.9	**24.3**	56.8	69.9	54.7	30.1	36.3	66.4	38.7	34.9	34.4	34.7	45.9
Strawberry	8.9	7.3	18.1	35.7	11.6	11.6	15.1	5.1	6.2	5.1	5.7	46.5	9.2	9.2	7.8	5.4	33.8	22.2	48.4	5.4	4.6	6.2	4.6	5.1	5.7	5.7	5.4	10.3
SwedishLeaf	28.8	11.7	27.0	70.9	26.6	26.6	28.0	21.1	22.2	19.0	13.0	86.6	22.7	22.7	38.6	21.1	63.7	45.4	74.7	22.6	**8.5**	12.8	44.2	25.9	12.8	15.0	12.3	44.5
Symbols	15.0	19.4	38.0	20.7	14.9	14.9	46.7	9.6	10.8	8.4	3.0	68.8	10.6	10.6	26.9	10.1	**0.0**	72.2	33.8	23.1	3.9	3.1	21.5	3.7	5.0	4.1	5.0	22.8
SyntheticControl	18.7	51.0	33.7	6.0	33.7	33.7	45.0	12.0	12.0	8.3	2.7	82.3	36.3	36.3	45.3	12.0	57.0	47.7	25.0	4.3	15.3	7.3	65.3	5.3	**0.3**	0.7	0.7	15.0
ToeSegmentation1	41.7	13.2	34.2	23.7	29.8	29.8	38.2	28.1	34.2	**10.1**	20.6	39.0	21.1	21.1	17.5	32.0	39.5	39.9	33.3	28.1	10.1	17.1	41.2	15.8	27.2	21.9	28.9	31.6
ToeSegmentation2	36.9	12.3	26.2	14.6	30.0	30.0	30.0	14.6	21.5	**7.7**	10.8	30.0	12.3	12.3	23.1	19.2	15.4	48.5	24.6	23.9	13.8	10.0	66.2	9.2	16.1	16.1	14.6	21.5
Trace	31.0	**0.0**	**0.0**	26.0	**0.0**	**0.0**	**0.0**	24.0	23.0	6.0	4.0	72.0	16.0	16.0	7.0	24.0	47.0	18.0	60.0	16.0	**0.0**	1.0	41.0	10.0	**0.0**	**0.0**	**0.0**	48.0
TwoLeadECG	27.1	7.1	31.0	48.5	23.1	23.1	22.7	26.5	25.6	2.2	3.2	19.4	21.2	21.2	27.0	25.3	37.7	36.3	44.2	28.8	**0.6**	7.8	15.3	8.4	9.6	9.6	9.6	27.5
TwoPatterns	79.6	54.9	71.1	0.1	66.8	66.8	67.1	3.9	27.1	5.5	0.4	74.8	50.0	50.0	71.3	9.3	66.2	73.5	28.9	37.9	0.1	0.0	61.3	0.5	**0.0**	**0.0**	**0.0**	40.4
UMD	38.2	25.0	27.8	12.5	29.9	29.9	44.4	13.2	25.0	13.2	8.3	28.5	31.9	31.9	41.7	23.6	54.9	50.0	41.7	16.7	**0.7**	4.9	22.2	9.7	**0.7**	**0.7**	**0.7**	27.8
UWaveGestureLibraryAll	13.3	39.1	71.9	4.0	71.6	71.6	82.0	4.6	5.9	5.1	5.6	77.6	30.2	30.2	71.9	5.2	60.8	83.2	5.9	11.0	5.8	4.6	58.5	12.3	5.3	8.3	5.3	6.9
UWaveGestureLibraryX	29.6	62.9	59.0	32.7	62.9	62.9	71.1	25.9	26.4	30.8	25.1	79.2	55.6	55.6	69.2	26.1	**0.0**	76.1	36.6	31.8	26.3	22.4	39.3	30.1	23.7	23.9	24.9	36.0
UWaveGestureLibraryY	37.4	68.6	59.0	40.8	60.6	60.6	67.1	33.4	34.2	33.4	33.7	82.0	64.5	64.5	66.0	33.8	**0.0**	73.0	47.6	40.3	35.8	30.9	58.3	39.1	30.7	33.3	31.2	46.0
UWaveGestureLibraryZ	38.8	**0.0**	**0.0**	37.5	**0.0**	**0.0**	**0.0**	35.0	35.1	33.9	32.7	81.2	**0.0**	55.5	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	38.7	33.8	31.4	44.8	34.5	31.6	32.5	33.1	**0.0**
Wafer	1.6	2.0	1.9	0.5	1.6	1.6	0.9	0.5	0.7	0.7	1.4	0.1	0.7	0.7	0.6	0.5	**0.0**	7.9	0.6	0.8	1.0	0.8	0.9	1.4	0.8	2.0	0.9	0.5
Wine	46.3	53.7	57.4	50.0	63.0	63.0	53.7	35.2	44.4	55.6	35.2	50.0	35.2	35.2	48.1	38.9	42.6	50.0	53.7	38.9	53.7	38.9	50.0	**24.1**	42.6	42.6	42.6	46.3
WordSynonyms	51.4	62.9	72.4	33.1	77.0	77.0	70.8	36.5	41.2	36.8	31.7	87.8	62.9	62.9	69.4	38.2	84.3	76.3	47.2	45.0	26.0	24.4	50.5	35.6	28.1	30.4	30.6	47.2
Worms	53.2	28.6	28.6	49.4	33.8	33.8	31.2	59.7	50.6	60.8	44.2	67.5	44.2	44.2	39.0	54.5	**0.0**	33.8	46.8	52.0	42.9	42.9	57.1	36.4	41.6	49.4	52.0	54.5
WormsTwoClass	37.7	22.1	23.4	40.3	26.0	26.0	20.8	42.9	32.5	34.3	39.0	40.3	32.5	32.5	36.4	39.0	**0.0**	24.7	33.8	36.4	27.3	33.8	39.0	31.2	39.0	40.3	36.4	37.7
Yoga	18.7	25.3	33.1	40.4	41.5	41.5	37.5	17.1	17.1	20.5	14.3	49.0	24.6	24.6	32.0	17.0	44.4	39.6	37.5	17.7	**11.7**	13.3	25.1	15.5	17.1	16.2	16.4	23.4
Number of Wins	1	5	4	1	3	3	2	1	1	10	4	0	4	3	2	2	13	1	1	2	24	13	3	2	9	9	8	2
Average rank	36.3	30.8	37.8	34.2	36.4	36.4	39.1	23.4	29.6	19.1	19.3	50.2	31.5	31.9	36.6	25.9	41.9	44.9	44.1	26.3	13.7	14.4	39.0	26.1	16.7	17.5	16.0	38.1
Mean error rate	37.9	33.5	40.9	40.3	40.7	40.7	43.2	29.1	32.5	24.6	25.5	61.3	33.0	33.5	40.2	30.3	49.3	50.6	48.6	30.0	21.5	23.3	43.2	28.7	24.6	24.8	24.6	39.0
Median error rate	35.0	32.3	39.2	37.5	40.0	40.0	42.5	25.9	30.1	21.7	23.8	65.3	31.7	31.9	38.6	26.7	49.0	46.2	46.5	28.1	18.0	21.7	41.9	25.8	22.0	21.1	21.6	38.3
Standard deviation of the error rate	22.7	20.7	22.0	24.4	21.7	21.7	22.7	20.7	20.9	18.8	18.0	22.5	18.5	18.3	22.3	20.6	24.3	21.5	22.9	19.4	17.9	18.5	24.0	19.6	18.6	19.0	18.8	20.2
Max error rate	94.2	85.4	89.2	99.0	86.2	86.2	90.0	94.7	94.7	88.2	83.2	98.1	87.4	87.4	92.3	94.2	93.5	95.4	97.4	96.2	73.6	86.5	97.1	79.3	89.9	90.4	89.9	94.9

CRediT authorship contribution statement

Tomasz Górecki: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing,. **Maciej Łuczak:** Software, Writing – original draft,. **Paweł Piasecki:** Data curation, Software, Writing – original draft, Funding acquisition,.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the people who contributed to the UCR time series repository. Resources supporting this work were provided by the Poznań Supercomputing and Networking Center.

References

[1] W. Dong, Q. Li, H.V. Zhao, Statistical and machine learning-based E-commerce sales forecasting, in: ICCSE'19: The 4th International Conference on Crowd Science and Engineering, Jinan, China, October 18-21, 2019, ACM, 2019, pp. 110–117, http://dx.doi.org/10.1145/3371238.3371256.

[2] J. Tang, X. Chen, Stock market prediction based on historic prices and news titles, in: Proceedings of the 2018 International Conference on Machine Learning Technologies, ICMLT '18, 2018, pp. 29–34, http://dx.doi.org/10.1145/3231884.3231887.

[3] G. Lucchese, W. Ketter, J. van Dalen, J. Collins, Forecasting prices in dynamic heterogeneous product markets using multivariate prediction methods, in: Proceedings of the 13th International Conference on Electronic Commerce, ICEC 2011, Liverpool, United Kingdom, August 3-5, 2011, ACM, 2011, pp. 26:1–26:10, http://dx.doi.org/10.1145/2378104.2378130.

[4] C. Yoo, J. Im, S. Park, L.J. Quackenbush, Estimation of daily maximum and minimum air temperatures in urban landscapes using MODIS time series satellite data, ISPRS J. Photogramm. Remote Sens. 137 (2018) 149–162, http://dx.doi.org/10.1016/j.isprsjprs.2018.01.018.

[5] T.T. Vu, J. Kiesel, B. Guse, N. Fohrer, Analysis of the occurrence, robustness and characteristics of abrupt changes in streamflow time series under future climate change, Clim. Risk Manage. 26 (2019) 100198, http://dx.doi.org/10.1016/j.crm.2019.100198.

[6] Z. Karevan, J.A.K. Suykens, Transductive LSTM for time-series prediction: An application to weather forecasting, Neural Netw. 125 (2020) 1–9, http://dx.doi.org/10.1016/J.NEUNET.2019.12.030.

[7] M.F. Iqbal, M. Zahid, D. Habib, L.K. John, Efficient prediction of network traffic for real-time applications, J. Comput. Netw. Commun. 2019 (2019) 4067135:1–4067135:11, http://dx.doi.org/10.1155/2019/4067135.

[8] M. Safaei, A.S. Ismail, H. Chizari, M. Driss, W. Boulila, S. Asadi, M. Safaei, Standalone noise and anomaly detection in wireless sensor networks: A novel time-series and adaptive Bayesian-network-based approach, Softw. - Pract. Exp. 50 (4) (2020) 428–446, http://dx.doi.org/10.1002/SPE.2785.

[9] R. Sipos, D. Fradkin, F. Mörchen, Z. Wang, Log-based predictive maintenance, in: S.A. Macskassy, C. Perlich, J. Leskovec, W. Wang, R. Ghani (Eds.), The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, ACM, 2014, pp. 1867–1876, http://dx.doi.org/10.1145/2623330.2623340.

[10] P. Montebruno, R.J. Bennett, H. Smith, C. van Lieshout, Machine learning classification of entrepreneurs in British historical census data, Inf. Process. Manage. 57 (3) (2020) 102210, http://dx.doi.org/10.1016/J.IPM.2020.102210.

[11] R. Rosas-Romero, E. Guevara, K. Peng, D.K. Nguyen, F. Lesage, P. Pouliot, W.-E. Lima-Saad, Prediction of epileptic seizures with convolutional neural networks and functional near-infrared spectroscopy signals, Comput. Biol. Med. 111 (2019) 103355, http://dx.doi.org/10.1016/j.compbiomed.2019.103355.

[12] L. Posthuma, C. Downey, M. Visscher, D. Ghazali, M. Joshi, H. Ashrafian, S. Khan, A. Darzi, J. Goldstone, B. Preckel, Remote wireless vital signs monitoring on the ward for early detection of deteriorating patients: A case series, International Journal of Nursing Studies 104 (2020) 103515, http://dx.doi.org/10.1016/j.ijnurstu.2019.103515.

[13] E. Fix, J. Hodges, Discriminatory Analysis : Nonparametric Discrimination, Consistency Properties, USAF School of Aviation Medicine, 1951.

[14] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27, http://dx.doi.org/10.1109/TIT.1967.1053964.

[15] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A.F.M. Ng, B. Liu, P.S. Yu, Z. Zhou, M.S. Steinbach, D.J. Hand, D. Steinberg, Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (1) (2008) 1–37, http://dx.doi.org/10.1007/S10115-007-0114-2.

[16] A. Abanda, U. Mori, J.A. Lozano, A review on distance based time series classification, Data Min. Knowl. Discov. 33 (2) (2019) 378–412, http://dx.doi.org/10.1007/S10618-018-0596-4.

[17] J. Serrà, M. Zanin, P. Herrera, X. Serra, Characterization and exploitation of community structure in cover song networks, Pattern Recognit. Lett. 33 (9) (2012) 1032–1041, http://dx.doi.org/10.1016/J.PATREC.2012.02.013.

[18] H.A. Dau, E. Keogh, K. Kamgar, C.-C.M. Yeh, Y. Zhu, S. Gharghabi, C.A. Ratanamahatana, C. Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, Hexagon-ML, The UCR time series classification archive, 2019.

[19] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, E.J. Keogh, Experimental comparison of representation methods and distance measures for time series data, Data Min. Knowl. Discov. 26 (2) (2013) 275–309, http://dx.doi.org/10.1007/S10618-012-0250-5.

[20] A.J. Bagnall, J. Lines, An experimental evaluation of nearest neighbour time series classification, 2014, CoRR abs/1406.4757.

[21] A.J. Bagnall, J. Lines, A. Bostrom, J. Large, E.J. Keogh, 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, http://dx.doi.org/10.1007/S10618-016-0483-9.

[22] T. Górecki, P. Piasecki, An experimental evaluation of time series classification using various distance measures, Arch. Data Sci. Ser. A 5 (1) (2018) 1–25, http://dx.doi.org/10.5445/KSP/1000087327/07.

[23] T. Górecki, P. Piasecki, A comprehensive comparison of distance measures for time series classification, in: A. Steland, E. Rafajłowicz, O. Okhrin (Eds.), Stochastic Models, Statistics and their Applications, Springer International Publishing, 2019, pp. 409–428, http://dx.doi.org/10.1007/978-3-030-28665-1_31.

[24] A. Dempster, D.F. Schmidt, G.I. Webb, MiniRocket: A very fast (almost) deterministic transform for time series classification, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 248–257, http://dx.doi.org/10.1145/3447548.3467231.

[25] C.W. Tan, A. Dempster, C. Bergmeir, G.I. Webb, MultiRocket: Multiple pooling operators and transformations for fast and effective time series classification, Data Min. Knowl. Discov. 36 (5) (2022) 1623–1646.

[26] A. Dempster, D.F. Schmidt, G.I. Webb, Hydra: Competing convolutional kernels for fast and accurate time series classification, Data Min. Knowl. Discov. 37 (5) (2023) 1779–1805, http://dx.doi.org/10.1007/s10618-023-00939-3.

[27] A. Bagnall, M. Flynn, J. Large, J. Lines, M. Middlehurst, On the usage and performance of the hierarchical vote collective of transformation-based ensembles version 1.0 (HIVE-COTE v1.0), in: Advanced Analytics and Learning on Temporal Data: 5th ECML PKDD Workshop, AALTD 2020, Ghent, Belgium, September 18, 2020, Revised Selected Papers, Springer-Verlag, Berlin, Heidelberg, 2020, pp. 3–18, http://dx.doi.org/10.1007/978-3-030-65742-0_1.

[28] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, A. Bagnall, HIVE-COTE 2.0: A new meta ensemble for time series classification, Mach. Learn. 110 (11–12) (2021) 3211–3243, http://dx.doi.org/10.1007/s10994-021-06057-9.

[29] A. Shifaz, C. Pelletier, F. Petitjean, G.I. Webb, TS-CHIEF: A scalable and accurate forest algorithm for time series classification, Data Min. Knowl. Discov. 34 (3) (2020) 742–775, http://dx.doi.org/10.1007/s10618-020-00679-8.

[30] M. Middlehurst, P. Schäfer, A. Bagnall, Bake off redux: a review and experimental evaluation of recent time series classification algorithms, 2023, arXiv:2304.13029.

[31] S. Alaee, A. Abdoli, C.R. Shelton, A.C. Murillo, A.C. Gerry, E.J. Keogh, Features or shape? Tackling the false dichotomy of time series classification, in: C. Demeniconi, N.V. Chawla (Eds.), Proceedings of the 2020 SIAM International Conference on Data Mining, SDM 2020, Cincinnati, Ohio, USA, May 7-9, 2020, SIAM, 2020, pp. 442–450, http://dx.doi.org/10.1137/1.9781611976236.50.

[32] P. Montero, J.A. Vilar, TSclust: An r package for time series clustering, J. Stat. Softw. 62 (1) (2014) 1–43.

[33] P. Esling, C. Agón, Time-series data mining, ACM Comput. Surv. Assoc. Comput. Mach. (CSUR) 45 (1) (2012) 12:1–12:34, http://dx.doi.org/10.1145/2379776.2379788.

[34] B. Yi, C. Faloutsos, Fast time sequence indexing for arbitrary L_p norms, in: A.E. Abbadi, M.L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, K. Whang (Eds.), VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt, Morgan Kaufmann, 2000, pp. 385–394.

[35] C. Antunes, A. Oliveira, Temporal data mining: An overview, in: Knowledge Discovery and Data Mining Workshop on Temporal Data Mining, 2001, pp. 1–13.

[36] B. Dhariyal, T. Le Nguyen, G. Ifrim, Back to basics: A sanity check on modern time series classification algorithms, in: G. Ifrim, R. Tavenard, A. Bagnall, P. Schaefer, S. Malinowski, T. Guyet, V. Lemaire (Eds.), Advanced Analytics and Learning on Temporal Data, Springer Nature Switzerland, Cham, 2023, pp. 205–229, http://dx.doi.org/10.1007/978-3-031-49896-1_14.

[37] E. Keogh, M. Pazzani, Dynamic time warping with higher order features, in: Proceedings of SIAM International Conference on Data Mining, Chicago, USA, 2001, pp. 1–11.

[38] C.S. Möller-Levet, F. Klawonn, K. Cho, O. Wolkenhauer, Fuzzy clustering of short time-series and unevenly distributed sampling points, in: M.R. Berthold, H. Lenz, E. Bradley, R. Kruse, C. Borgelt (Eds.), Advances in Intelligent Data Analysis V, 5th International Symposium on Intelligent Data Analysis, IDA 2003, Berlin, Germany, August 28-30, 2003, Proceedings, in: Lecture Notes in Computer Science, vol. 2810, Springer, 2003, pp. 330–340, http://dx.doi.org/10.1007/978-3-540-45231-7_31.

[39] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: U.M. Fayyad, R. Uthurusamy (Eds.), Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop, Seattle, Washington, USA, July 1994. Technical Report WS-94-03, AAAI Press, 1994, pp. 359–370.

[40] E.J. Keogh, C.A. Ratanamahatana, Exact indexing of dynamic time warping, Knowl. Inf. Syst. 7 (3) (2005) 358–386, http://dx.doi.org/10.1007/S10115-004-0154-9.

[41] U. Mori, A. Mendiburu, J.A. Lozano, Distance measures for time series in R: The TSdist package, R J. 8 (2) (2016) 451–459, http://dx.doi.org/10.32614/RJ-2016-058.

[42] T. Górecki, M. Łuczak, First and second derivatives in time series classification using DTW, Comm. Statist. Simulation Comput. 43 (9) (2014) 2081–2092, http://dx.doi.org/10.1080/03610918.2013.775296.

[43] M. Łuczak, Univariate and multivariate time series classification with parametric integral dynamic time warping, J. Intell. Fuzzy Systems 33 (4) (2017) 2403–2413, http://dx.doi.org/10.3233/JIFS-17523.

[44] J. Zhao, L. Itti, ShapeDTW: Shape dynamic time warping, Pattern Recognit. 74 (2018) 171–184, http://dx.doi.org/10.1016/J.PATCOG.2017.09.020.

[45] G.E.A.P.A. Batista, X. Wang, E.J. Keogh, A complexity-invariant distance measure for time series, in: Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28-30, 2011, Mesa, Arizona, USA, SIAM / Omni Press, 2011, pp. 699–710, http://dx.doi.org/10.1137/1.9781611972818.60.

[46] S. Gharghabi, S. Imani, A.J. Bagnall, A. Darvishzadeh, E.J. Keogh, Matrix profile XII: MPdist: A novel time series distance measure to allow data mining in more challenging scenarios, in: IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018, IEEE Computer Society, 2018, pp. 965–970, http://dx.doi.org/10.1109/ICDM.2018.00119.

[47] C.M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H.A. Dau, D.F. Silva, A. Mueen, E.J. Keogh, Matrix profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets, in: F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z. Zhou, X. Wu (Eds.), IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain, IEEE Computer Society, 2016, pp. 1317–1322, http://dx.doi.org/10.1109/ICDM.2016.0179.

[48] D. Folgado, M. Barandas, R. Matias, R. Martins, M. Carvalho, H. Gamboa, Time alignment measurement for time series, Pattern Recognit. 81 (2018) 268–279, http://dx.doi.org/10.1016/j.patcog.2018.04.003.

[49] M. Vlachos, D. Gunopulos, G. Kollios, Discovering similar multidimensional trajectories, in: R. Agrawal, K.R. Dittrich (Eds.), Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002, IEEE Computer Society, 2002, pp. 673–684, http://dx.doi.org/10.1109/ICDE.2002.994784.

[50] T. Górecki, Classification of time series using combination of DTW and LCSS dissimilarity measures, Comm. Statist. Simulation Comput. 47 (1) (2018) 263–276, http://dx.doi.org/10.1080/03610918.2017.1280829.

[51] C.A. Ratanamahatana, J. Lin, D. Gunopulos, E. Keogh, M. Vlachos, G. Das, Mining time series data, in: Data Mining and Knowledge Discovery Handbook, Springer US, Boston, MA, 2010, pp. 1049–1077.

[52] T. Górecki, Using derivatives in a longest common subsequence dissimilarity measure for time series classification, Pattern Recognit. Lett. 45 (2014) 99–105, http://dx.doi.org/10.1016/J.PATREC.2014.03.009.

[53] L. Chen, M.T. Özsu, V. Oria, Robust and fast similarity search for moving object trajectories, in: F. Özcan (Ed.), Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005, ACM, 2005, pp. 491–502, http://dx.doi.org/10.1145/1066157.1066213.

[54] L. Chen, R.T. Ng, On the marriage of Lp-norms and edit distance, in: M.A. Nascimento, M.T. Özsu, D. Kossmann, R.J. Miller, J.A. Blakeley, K.B. Schiefer (Eds.), (E)Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, Toronto, Canada, August 31 - September 3 2004, Morgan Kaufmann, 2004, pp. 792–803, http://dx.doi.org/10.1016/B978-012088469-8.50070-X.

[55] A. Stefan, V. Athitsos, G. Das, The move-split-merge metric for time series, IEEE Trans. Knowl. Data Eng. 25 (6) (2013) 1425–1438, http://dx.doi.org/10.1109/TKDE.2012.88.

[56] X. Golay, S. Kollias, G. Stoll, D. Meier, A. Valavanis, P. Boesiger, A new correlation-based fuzzy logic clustering algorithm for fMRI, Magn. Reson. Med. 40 (2) (1998) 249–260.

[57] T.W. Liao, Clustering of time series data - a survey, Pattern Recognit. 38 (11) (2005) 1857–1874, http://dx.doi.org/10.1016/J.PATCOG.2005.01.025.

[58] D. Peña, P. Galeano, Multivariate Analysis in Vector Time Series, Technical Report ws012415, Universidad Carlos III de Madrid. Departamento de Estadística, 2001, URL https://ideas.repec.org/p/cte/wsrepe/ws012415.html.

[59] A.D. Chouakria, P.N. Nagabhushan, Adaptive dissimilarity index for measuring time series proximity, Adv. Data Anal. Classif. 1 (1) (2007) 5–21, http://dx.doi.org/10.1007/S11634-006-0004-6.

[60] R. Agrawal, C. Faloutsos, A.N. Swami, Efficient similarity search in sequence databases, in: D.B. Lomet (Ed.), Foundations of Data Organization and Algorithms, 4th International Conference, FODO'93, Chicago, Illinois, USA, October 13-15, 1993, Proceedings, in: Lecture Notes in Computer Science, vol. 730, Springer, 1993, pp. 69–84, http://dx.doi.org/10.1007/3-540-57301-1_5.

[61] J. Aßfalg, H. Kriegel, P. Kröger, P. Kunath, A. Pryakhin, M. Renz, Similarity search on time series based on threshold queries, in: Y.E. Ioannidis, M.H. Scholl, J.W. Schmidt, F. Matthes, M. Hatzopoulos, K. Böhm, A. Kemper, T. Grust, C. Böhm (Eds.), Advances in Database Technology - EDBT 2006, 10th International Conference on Extending Database Technology, Munich, Germany, March 26-31, 2006, Proceedings, in: Lecture Notes in Computer Science, vol. 3896, Springer, 2006, pp. 276–294, http://dx.doi.org/10.1007/11687238_19.

[62] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E.J. Keogh, Querying and mining of time series data: experimental comparison of representations and distance measures, Proc. VLDB Endow. 1 (2) (2008) 1542–1552, http://dx.doi.org/10.14778/1454159.1454226.

[63] J. Caiado, N. Crato, D. Peña, A periodogram-based metric for time series classification, Comput. Statist. Data Anal. 50 (10) (2006) 2668–2684, http://dx.doi.org/10.1016/J.CSDA.2005.04.012.

[64] D. Casado de Lucas, Classification Techniques for Time Series and Functional Data (Ph.D. thesis), Universidad Carlos III de Madrid, 2010.

[65] J. Fan, I. Gijbels, Local Polynomial Modelling and its Applications, in: Monographs on Statistics and Applied Probability Series, vol. 66, Chapman & Hall, London, 1996, http://dx.doi.org/10.1201/9780203748725.

[66] Y. Kakizawa, R.H. Shumway, M. Taniguchi, Discrimination and clustering for multivariate time series, J. Amer. Statist. Assoc. (ISSN: 01621459) 93 (441) (1998) 328–340.

[67] S.P. Díaz, J.A. Vilar, Comparing several parametric and nonparametric approaches to time series clustering: A simulation study, J. Classification 27 (3) (2010) 333–362, http://dx.doi.org/10.1007/S00357-010-9064-6.

[68] J. Lin, E.J. Keogh, S. Lonardi, B.Y. Chiu, A symbolic representation of time series, with implications for streaming algorithms, in: M.J. Zaki, C.C. Aggarwal (Eds.), Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD 2003, San Diego, California, USA, June 13, 2003, ACM, 2003, pp. 2–11, http://dx.doi.org/10.1145/882082.882086.

[69] E. Keogh, The SAX (symbolic aggregate aproximation), 2023, http://www.cs.ucr.edu/~eamonn/SAX.htm, Accessed: 2023-25-12.

[70] J. Lin, E.J. Keogh, L. Wei, S. Lonardi, Experiencing SAX: a novel symbolic representation of time series, Data Min. Knowl. Discov. 15 (2) (2007) 107–144, http://dx.doi.org/10.1007/S10618-007-0064-Z.

[71] D. Piccolo, A distance measure for classifying ARIMA models (Corr: V11 p180), J. Time Series Anal. 11 (2) (1990) 153–164, http://dx.doi.org/10.1111/j.1467-9892.1990.tb00048.x.

[72] E.J. Keogh, S. Lonardi, C.A. Ratanamahatana, Towards parameter-free data mining, in: W. Kim, R. Kohavi, J. Gehrke, W. DuMouchel (Eds.), Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004, ACM, 2004, pp. 206–215, http://dx.doi.org/10.1145/1014052.1014077.

[73] Z. Jiang, M. Yang, M. Tsirlin, R. Tang, Y. Dai, J. Lin, "Low-resource" text classification: A parameter-free classification method with compressors, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 6810–6828, http://dx.doi.org/10.18653/v1/2023.findings-acl.426.

[74] A. Brandmaier, Permutation Distribution Clustering and Structural Equation Model Trees (Ph.D. thesis), Saarland University, 2012.

[75] T. Górecki, M. Łuczak, Using derivatives in time series classification, Data Min. Knowl. Discov. 26 (2) (2013) 310–331, http://dx.doi.org/10.1007/S10618-012-0251-4.

[76] T. Górecki, M. Łuczak, Non-isometric transforms in time series classification using DTW, Knowl.-Based Syst. 61 (2014) 98–108, http://dx.doi.org/10.1016/J.KNOSYS.2014.02.011.

[77] A. Bagnall, J. Lines, J. Hills, A. Bostrom, Time-series classification with COTE: The collective of transformation-based ensembles, IEEE Trans. Knowl. Data Eng. 27 (9) (2015) 2522–2535, http://dx.doi.org/10.1109/TKDE.2015.2416723.

[78] J. Lines, A. Bagnall, Time series classification with ensembles of elastic distance measures, Data Min. Knowl. Discov. 29 (3) (2015) 565–592, http://dx.doi.org/10.1007/s10618-014-0361-2.

[79] H.A. Dau, A. Bagnall, K. Kamgar, C.-C.M. Yeh, Y. Zhu, S. Gharghabi, C.A. Ratanamahatana, E. Keogh, The UCR time series archive, 2019, arXiv:1810.07758.

[80] E.J. Keogh, S. Kasetty, On the need for time series data mining benchmarks: A survey and empirical demonstration, Data Min. Knowl. Discov. 7 (4) (2003) 349–371, http://dx.doi.org/10.1023/A:1024988512476.

[81] R.C. Team, R: A language and environment for statistical computing, 2022, URL https://www.R-project.org/.

[82] F. Bischoff, Tsmp: Time series with matrix profile, 2022, URL https://CRAN.R-project.org/package=tsmp, R package version 0.4.15.

[83] B. Calvo, G. Santafé, Scmamp: Statistical comparison of multiple algorithms in multiple problems, R J. 8 (1) (2016) 248, http://dx.doi.org/10.32614/RJ-2016-017.

[84] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Amer. Statist. Assoc. 32 (200) (1937) 675–701, http://dx.doi.org/10.2307/2279372.

[85] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1) (1940) 86–92.

[86] R.L. Iman, J.M. Davenport, Approximations of the critical region of the Friedman statistic, Commun. Stat. – Theory Methods 9 (6) (1980) 571–595, http://dx.doi.org/10.1080/03610928008827904.

[87] S. García, F. Herrera, An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (12) (2008) 2677–2694.

[88] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Inform. Sci. 180 (10) (2010) 2044–2064, http://dx.doi.org/10.1016/J.INS.2009.12.010.

[89] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[90] P. Nemenyi, Distribution-Free Multiple Comparisons (Ph.D. thesis), Princeton University, 1963.

[91] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (6) (1945) 80–83, http://dx.doi.org/10.2307/3001968.

[92] B. Bergmann, G. Hommel, Improvements of general multiple test procedures for redundant systems of hypotheses, in: P. Bauer, G. Hommel, E. Sonnemann (Eds.), Multiple HypothesenprÜFung / Multiple Hypotheses Testing, Springer Berlin Heidelberg, Berlin, Heidelberg, 1988, pp. 100–115, http://dx.doi.org/10.1007/978-3-642-52307-6_8.

[93] C. Spearman, The proof and measurement of association between two things, Amer. J. Psychol. 15 (1) (1904) 72–101, http://dx.doi.org/10.2307/1412159.

Tomasz Górecki, Ph.D., D.Sc. is an Associate Professor at the Department of Mathematical Statistics and Data Analysis, Faculty of Mathematics and Computer Science, Adam Mickiewicz University in Poznań. His main research interests include methods of artificial intelligence, machine learning, and time series analysis and their applications. He has many years of experience in cooperation with industry, including Lidl and Samsung, where he worked, among others, on-demand and inventory forecasting systems and intelligent voice systems. In addition, for many years he has been cooperating with specialists in other fields of science, such as economics, chemistry, geography, or transport, combining practice with theory. He is the author of over 100 scientific papers and 3 books.

Maciej Łuczak, Ph.D., D.Sc. is an Associate Professor at the Department of Mathematical Statistics and Data Analysis, Faculty of Mathematics and Computer Science, Adam Mickiewicz University in Poznań. His main research interests include methods of machine learning, and time series analysis. He is the author of over 20 scientific papers.

Paweł Piasecki, Ph.D. is a Senior Data Scientist at Scalo. He is the author of 4 scientific papers.