Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics en Operations Research

Determining the Optimal Number of Clusters in Time Series Clustering Using Elastic Distance Functions

Luc Braks (588744)

Supervisor:	drs. J. Durieux
Second assessor:	M.F.O. Welz
Date final version:	1st July 2024

Abstract

The growing interest in clustering time series data across various disciplines has spurred research into innovative methods. The majority of time series clustering techniques utilize traditional clustering algorithms like k-means and k-medoids, which require the number of clusters k to be known prior to clustering. This study focuses on determining the number of clusters in a time series dataset using three model selection methods: silhouette, Davies-Bouldin (DB) and Calinski-Harabasz (CH). This analysis is conducted in combination with partitional clustering algorithms that employ elastic distance measures. This study commences with an in-depth analysis on five elastic distance functions using the UCR time series archive. The findings of this study indicate that k-medoids combined with the time warp edit (TWE) distance proved to be the most effective algorithm, followed closely by the move-split-merge (MSM) and edit distance with real penalty (ERP). In determining the optimal of clusters within a time series dataset, the silhouette method was found to be the most effective model selection method, followed by the CH, with the DB performing significantly worse. The silhouette method combined with the TWE and MSM was best able to replicate the number of clusters as defined in the UCR archive. Both MSM and ERP demonstrated the best performance in defining well-separated clusters, reaffirming the competitive advantage of elastic distance functions in time series clustering.

1 Introduction

Clustering is the process of grouping data into distinct groups based on their patterns, such that data with similar patterns are clustered together, whereas data with dissimilar patterns are kept separate and is essential in various interdisciplinary research areas, including mathematics, statistics, finance, and biology (Rokach and Maimon, 2005). Time series clustering (TSCL) is a type of clustering, which involves the categorization of ordered time series data without relying on a label. As the utilization of time series data continues to grow in both academic and real-world applications, the significance of TSCL research has surged. Generally, there are two main methods for TSCL. The first approach involves extracting features from the time series data and the second approach uses the raw time series, employing a distance measure between the series to execute the clustering procedure. This research focuses on the analysis of raw time series data.

Researchers have shown significant interest in TSCL, leading to the development of several elastic distance measures over the past decades. There have been numerous comparative studies which use the time series databases of the University of California, Riverside (UCR) archive (Dau et al., 2019), assessing their performance on how well they recreate class labels from the UCR databases (Aghabozorgi et al., 2015; Javed et al., 2020). In particular, Holder et al. (2024) conducted a comprehensive review of partitional clustering algorithms that use elastic distance measures. However, all the algorithms they employ require the number of clusters, denoted as k, to be known prior to clustering. A noteworthy quote from them is: "If a clustering simply finds what we already know, its utility is limited". One could argue that if the number of clusters is known prior to clustering, the value of such algorithms is equally restricted. This study builds on their work by investigating the scenario where the cluster count is unknown prior to clustering, which holds substantial significance in the field of time series clustering.

In their study, Holder et al. (2024) employ ten different distance measures that are commonly used in TSCL literature. They make a detailed empirical analysis using the univariate time series data of the UCR archive. The authors note that the k-means algorithm, also known as Lloyd's algorithm (Lloyd, 1982), is used predominately in TSCL literature, whereas the application of k-medoids clustering (Rdusseeun and Kaufman, 1987) is less common. In their study, they utilize partition based clustering algorithms for TSCL, where they set the cluster count to the pre-determined number of classes of the UCR time series databases.

Holder et al. (2024) uncover a series of interesting findings that shed new light on the subject of TSCL. They conclude that when applying k-means clustering, only one elastic distance measure, move-split-merge (MSM), out of the nine investigated performs significantly better than Euclidean distance (ED). In contrast, five of the nine measures, including dynamic time warping (DTW), perform significantly worse. When the analysis is repeated using the k-medoids algorithm an increase in clustering performance is found for all investigated distances except ED. With k-medoids clustering, DTW is no longer significantly worse than ED. With both investigated algorithms, the move-split-merge distance function emerged as the best performing method, closely followed by time-warp-edit (TWE) distance. Finally, they conclude that altering the tuning parameters for both distance functions and algorithms did not enhance clustering performance.

This study commences with a replication of the study by Holder et al. (2024), to verify their claimed advantage of elastic distance functions. Additionally, the study compares the performance of the k-means and k-medoids clustering algorithms to confirm the claimed advantage of k-medoids clustering.

A key aim of this study is to challenge the traditional assumption in TSCL of a pre-determined number of clusters. It is of high importance to TSCL practitioners, who often deal with unclassified or not yet classified time series data. Therefore, this study investigates whether the previously reported performance of the MSM and TWE distances measures remains consistent without a pre-determined number of clusters. To achieve this, three model selection methods are employed. The research objective is twofold: first, to evaluate the accuracy of the cluster count against the pre-defined number in the UCR archive, and second, to identify which elastic distance measure can best group similar data points together and separated dissimilar data points. Hence, the primary research question is: *How does the need to pre-determine the number of clusters impact the performance of elastic distance measures in time series clustering?* This question is of high importance to future research on TSCL, as no such research has been conducted before, and it will provide TSCL practitioners with a method for handling unclassified time series data.

Firstly, this research confirms the earlier established superiority of the k-medoids clusterer over the k-means clusterer, as reported by Holder et al. (2024). These findings once again confirm that the k-medoids clusterer should be seen as the benchmark algorithm for TSCL, despite the fact that TSCL practitioners commonly use the k-means clustering as a benchmark. With both k-means and k-medoids clustering the TWE distance function emerged as the best performing distance, with the MSM a close second.

For the analysis on determining the optimal number of clusters in a dataset three model

selection methods are utilized: silhouette index (Rousseeuw, 1987), Davies-Bouldin (DB) index (Davies and Bouldin, 1979) and Calinski-Harabasz (CH) index (Caliński and Harabasz, 1974). The predicted number of clusters was then compared to the "true" number of clusters in each dataset, as declared in the UCR archive. The silhouette method combined with TWE and MSM emerged as the best performing methods. The DB method performs worse, and as such it is not recommended for TSCL practitioners to use this method. Silhouette and CH indices perform well when the cluster count is low. However, with increasing cluster count, all model selection methods perform poorly. As such, this study would advise against using these three model selection methods when TSCL practitioners want to cluster data in a large number of cluster.

This analysis on model selection assumes that the UCR archive's number of clusters is correct. However, there is skepticism about whether the UCR archive accurately represents real-world TSCL problems, as having many clusters in a dataset seems unreasonable. Therefore, clustering analysis is performed with the predicted number of clusters by the silhouette index, therefore not relying on the "true" number of clusters of the UCR archive. These clusters are consequently evaluated against each other using the Rand index and Mutual information score. MSM and ERP emerge as the best methods for clustering the time series, indicating that they produce the best defined clusters in which time series in different clusters are well separated, while those in the same cluster are similar.

The remainder of this paper is structured in the following manner. An overview of relevant literature is presented in Section 2. Section 3 describes the data used in this study. In Section 4, the notation is introduced and it provides a detailed explanation of all methods used. Finally, the conclusion of the research is presented in Section 6, with suggestions for further research.

2 Literature Review

This study contributes to the vast literature on time series clustering. TSCL generally consists of two approaches: using feature extraction and using raw series. Feature-based clustering works by converting raw series into feature vectors with lower dimension and cluster the series based on these vectors. Räsänen and Kolehmainen (2009) extract statistical feature (mean, standard deviation, skewness etc.) from electricity usage time series and cluster the series using k-means clustering. Also more complex unsupervised feature extraction algorithms exist, like the method proposed by Zhang et al. (2006), where they use the orthogonal wavelet transform to perform the dimension reduction.

This study emphasis on the raw data TSCL approach, using partition based clustering methods that utilize elastic distances. Clustering time series based on raw data often uses partition-based clustering algorithms, like the k-means algorithm originally proposed by Lloyd (1982). This algorithm generates k centroids and then clusters the data around those centroids using a distance measure. The two most widely researched and used distance measures in TSCL are the Euclidean distance (Faloutsos et al., 1994) and the dynamic time warping distance (Berndt and Clifford, 1994). A comprehensive review on elastic distance based clustering is provided by Holder et al. (2024), where they compare the performance between k-means and k-medoids clustering using ten different distance functions. Other partition-based clustering algorithms for TSCL include the k-shape algorithm (Paparrizos and Gravano, 2015), which uses

the cross correlation to compute the similarity between the time series.

More recent, the use of deep learning is developed for the use in TSCL, with Lafabregue et al. (2022) providing a comprehensive review of deep learning TSCL methods.

One problem with these partition based clustering algorithms, is that they require the number of cluster, k, to be known prior to clustering. In literature there seems to be no agreement on a method that optimally determines the number of clusters in a dataset. Milligan and Cooper (1985) conclude that the optimal method depends highly on the type of data used and the context of the problem. One common method of determining the number of clusters is the elbow method. However, this method is highly subjective and when the curve of explained variance is smooth, it is difficult to determine the elbow's location (Kanakarathinam, 2017). As such, more quantitative method have been proposed. Such model selection methods include the Silhouette method (Rousseeuw, 1987), Davies-Bouldin score (Davies and Bouldin, 1979), Calinski-harabasz index (Caliński and Harabasz, 1974) and the Gap statistic (Tibshirani et al., 2001). All of these methods aim to identify the value of k that results in the best-clustered data.

Even though the literature on how to determine the optimal number of clusters in a dataset is vast, the use of these methods in TSCL remains scarce. Raihan (2023) performs a model selection analysis, utilizing both raw data and a form of feature extraction. However, when using raw data, the study relied solely on the silhouette method and using the Euclidean distance to determine the optimal number of clusters. They conclude that feature extraction methods firmly outperform raw data methods.

3 Data

In this section, the data selected for the research is discussed. The time series data in the University of California, Riverside (UCR) archive (Dau et al., 2019) is utilized, with a focus on the univariate time series databases of the archive[1]. The UCR is the golden standard in time series clustering analysis and used often in TSCL literature (Javed et al., 2020; Ma et al., 2019). The total univariate UCR archive consists of 128 datasets. This research excludes the same datasets as Holder et al. (2024). This involves excluding time series of unequal lengths, series with missing values and the exclusion of the "Fungi" dataset, as this dataset contains only one training observation per cluster. With these restrictions, 112 datasets remain to be used for the research. This dataset is the same used by Holder et al. (2024).

The range of the number of cluster in the UCR archive ranges from 2 to 60 cluster. Holder et al. (2024) divide these datasets in four groups depending on their number of clusters: Group A: 2 clusters; Group B: 3-5 clusters; Group C: 6-10 clusters and Group D: 11 or more clusters. These groups contain 40, 33, 19 and 13 datasets respectively. This is to provide an extensive comparison between clustering problems of with varying cluster counts.

One important issue with clustering algorithms is that they are computationally intensive. Given this, decisions are made about which problems to address. This research utilizes a subset of the 112 available datasets. Clustering problems with large number of classes, series with long series length and series with large train sets are the primary contributors to the increased

[1]The UCR archive can be retrieved from timeseriesclassification.com/

computationally intensity. It is crucial to investigate datasets from of all of the mentioned groups. Group D only contains 13 datasets, but they have long series length and large training sets, making them computationally the most intensive problems. The computation of datasets in Group D, however, comes at the expense of replicating all the distance functions by Holder et al. (2024). However, this approach is necessary to allow for a comprehensive study on the model selection methods. The full subset of datasets used in this research is listed in Table 2 in Appendix A.

The UCR data presents several challenge for the analysis of clustering algorithms, as mentioned by Hu et al. (2016). Significant portion of the data sets have been subjected to preprocessing with expert knowledge, which may give biases to certain time series or cluster. In certain instances, the division of train/test sets was performed manually, which in turn could also result in a clustering bias. Despite the mentioned shortcomings, the datasets contained in the UCR archive are the state-of-the-art datasets for evaluating new TSCL methods and comparing existing methods. Therefore, these limitations should not hinder the research.

4 Methodology

In this section the methods used in this study are discussed. The methods can be categorized into four distinct groups. In Section 4.1, the notation for the replication of the study by Holder et al. (2024) is given, where the underlying principles of the utilized distance functions are discussed. Secondly, Section 4.2 presents the performance measures used to assess the clustering performance of the various distance functions. In Section 4.3, model selection methods are presented, followed by the evaluation of these methods in Section 4.4.

4.1 Distance functions

This section offers an extensive overview on the time series distance measures used in the study. These measures include the Euclidean distance and five elastic distance measures. The notation in this section mostly follows that of Holder et al. (2024). Certain elastic distance functions contain parameters, and as advised by Holder et al., these are all assigned their default values.

4.1.1 Euclidean distance

Given two time series of equal length, $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, the Euclidean distance (ED) is defined as

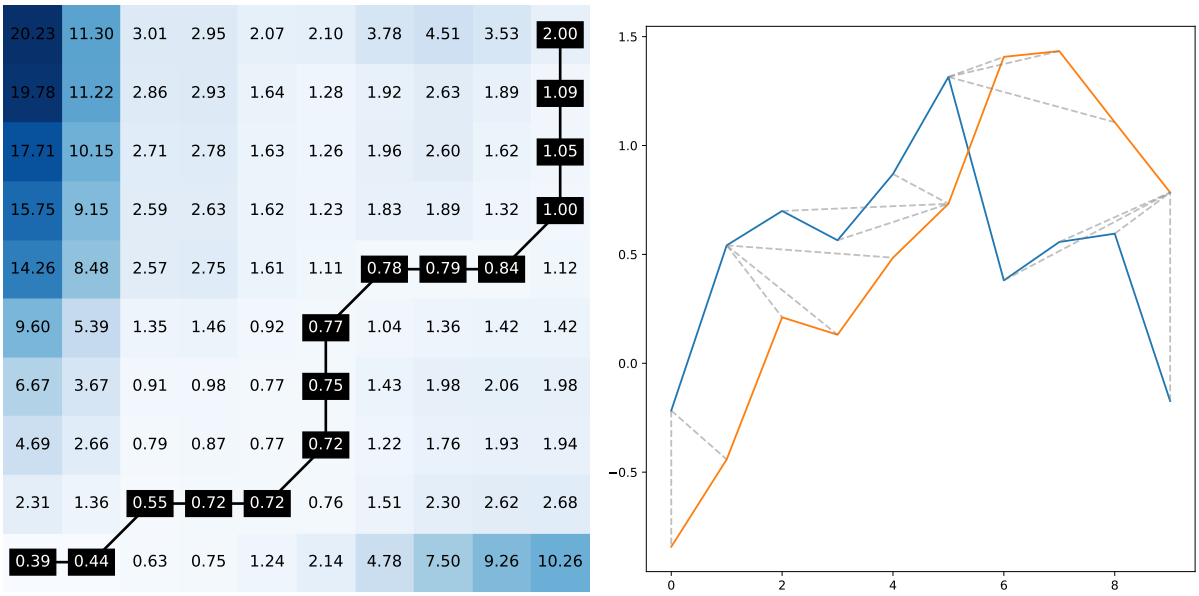
$$d_{\text{ed}}(A, B) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}. \tag{1}$$

The ED is arguably the simplest distance measure between two time series and its efficiency and ease of use have made it one of the most popular distance measures in TSCL (Aghabozorgi et al., 2015). However, the Euclidean distance suffers from a big disadvantage: it is very sensitive to small offsets of two time series. For instance, if two identical time series are slightly shifted relative to each other, the ED would then give an excessively large distance between the two

series, even though they are very similar. This disadvantage of the ED can be overcome using elastic distance measures.

4.1.2 Dynamic time warping

Dynamic time warping (DTW) is the most commonly used elastic distance measure in TSCL literature. It was introduced to time series problems by Berndt and Clifford (1994) and is a method which realigns time series to achieve the best match between them, making it more robust to small offsets compared to the ED. The first step in aligning the two time series involves constructing the cost matrix M, in which element $M_{i,j} = (a_i - b_j)^2$. $M_{i,j}$ represents the cost to align point a_i of time series A to the point b_j of time series B. Essentially, M is an $m \times m$ matrix that captures the pointwise distances between series A and B.



(a) Cost matrix C

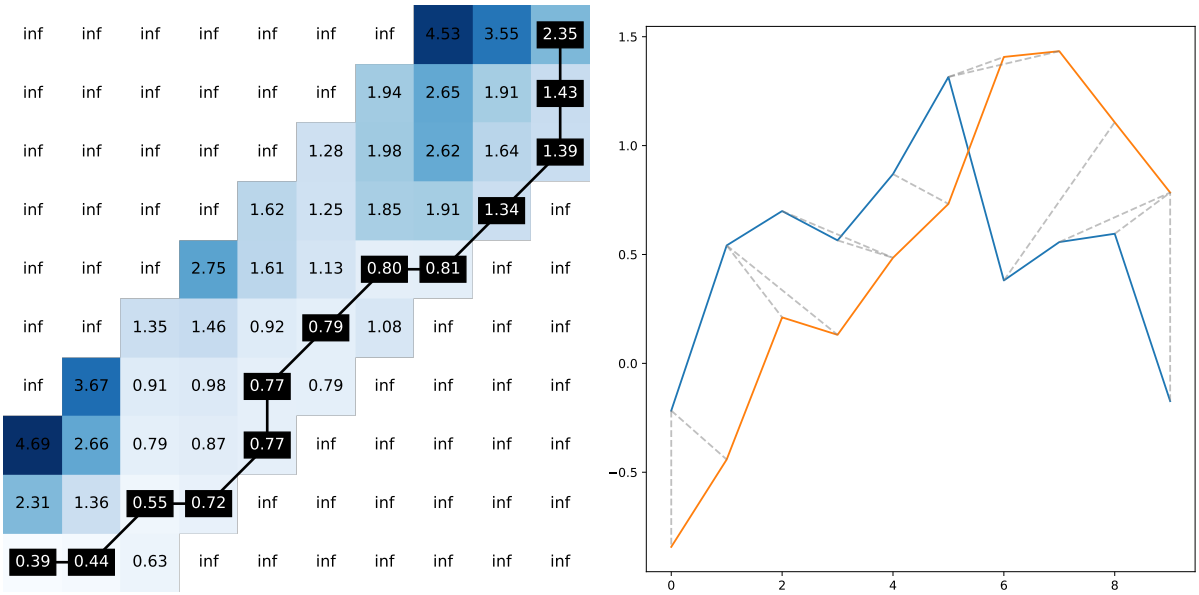
(b) DTW alignment

Figure 1: An example of the DTW distance for two simulated time series. Left figure (a) shows cost matrix M and right figure (b) shows the two time series with optimal alignment path.

An alignment between the two time series can be thought of as a warping path through the cost matrix M, which starts at the bottom-left corner and ends at the upper-right corner. For a warping path to be valid, it must not track-back. The dynamic time warping distance between two time series is determined by the optimal warping path P^*, trough the cost matrix M that minimizes the total cost. Figure 1 shows a practical demonstration of the DTW procedure on two simulated time series[2].

The optimal path P^* can be computed using an efficient dynamic programming algorithm. This involves creating a new $m \times m$ matrix C. The algorithm is initialised at (0,0), where $C_{0,0}$ is equal to $M_{0,0}$. The rest of the matrix is then filled recursively by calculating the distance between corresponding points of the two time series and adding it to the minimum of the three adjacent cells. Mathematically, this can be expressed as: $C_{i,j} = M_{i,j} + \min(C_{i-1,j}, C_{i,j-1}, C_{i-1,j-1})$.

[2]Code for the simulation can be found on github.com/luckyluc99/TSCL-model-selection



(a) Cost matrix C, with Sakoe and Chiba band

(b) Constrained DTW alignment ($w_{sc} = 0.2$)

Figure 2: An example of the DTW distance with Sakoe-Chiba band ($w_{sc} = 0.2$), for two simulated time series. Left figure (a) shows cost matrix M and right figure (b) shows the two time series with optimal alignment path.

This algorithm is however a computationally intensive task. As such, several methods have been proposed to improve the efficiency of the algorithm (Itakura, 1975; Sakoe and Chiba, 1978). Both of these methods impose constraints on the possible alignment paths between the time series. In this study, the slope constraint by Sakoe and Chiba is utilised. The constraint is set up symmetrically around the diagonal of the M matrix, such that no path can occur outside the boundaries. Added benefit of this method is that the path is not able to drift very far from the diagonal, which could improve accuracy (Ratanamahatana and Keogh, 2004).

Practically, the Sakoe-Chiba band works by setting values of the matrix M outside of the band to infinity. These constraining bands are applied to the simulation example of Figure 1 and can be seen in Figure 2, where the bounding window is set to 20% ($w_{sc} = 0.2$) of the time series. The dynamic programming algorithm with Sakoe-Chiba constraint is presented in Algorithm 1 in Appendix B.

4.1.3 Weighted dynamic time warping

The weighted dynamic time warping (WDTW) distance, proposed by Jeong et al. (2011), is a penalty-based DTW. The method was first proposed as an approach to prevent minimum distance distortion caused by outliers.

The process of WDTW functions by adding a weight penalty w to all elements of the pointwise distance matrix, M. This penalty depends on the warping distance $|i - j|$, resulting in the formation of the weighted pointwise matrix M^w. These elements are the product of the weight penalty and the pointwise distance, such that $M^w_{i,j} = w(|i - j|) \cdot M_{i,j}$.

Jeong et al. (2011) propose a modified logistic weight function, whereby a warping of p places

results in a penalty weight of:

$$w(p) = \frac{w_{max}}{1 + \exp(-g(p - m_c))} \tag{2}$$

where m_c is the midpoint of the time series, and g is the parameter that controls the penalty level for large warpings. In this study, the default value $g = 0.05$ is selected. Note that WDTW uses the same full window as that of the unconstrainted DTW, which can be computationally intensive. Algorithm 1 can be used to compute the WDTW distance between two time series, with inputs $M = M^w$ and $w_{sc} = 1$.

4.1.4 Edit distance with real penalty

The edit distance with real penalty (ERP) (Chen and Ng, 2004), is a distance measure which combines the L1 (Manhatten) distance and the use of edit operations (insertion, deletion and substitution). The goal of ERP is to transform one time series into another using these edit operations. The ERP is similar to DTW, as both describe an alignment path between the two series. However, the ERP uses the Manhatten distance rather than the Euclidean distance. This ensures that $d(a_i, b_j) = |a_i - b_j|$ is the cost of substitution between two points. Additionally, the costs of deletion/insertion are defined as $d(a_i, g) = |a_i - g|$ and $d(b_i, g) = |b_i - g|$. In these formulas, g is the penalty value, and is set to the default 0.05.

The ERP uses a cost matrix E where $E_{i,j}$ denotes the minimum cost of transforming the first i points of series A into the first j points of time series B. To do this, first the edges of the matrix E are initialized: $E_{i,0} = \sum_{k=1}^{i} d(a_i, g) = \sum_{k=1}^{i} |a_i - g|$ and $E_{0,j} = \sum_{k=1}^{j} d(b_i, g) = \sum_{k=1}^{j} |b_j - g|$ The rest of the matrix is then filled recursively by finding the minimum cost to get to that point. This is done using the following criterion:

$$E_{i,j} = \min \begin{cases} E_{i-1,j} + |a_i - g|, \\ E_{i,j-1} + |b_j - g|, \\ E_{i-1,j-1} + |a_i - b_j| \end{cases} \tag{3}$$

The ERP distance between the time series A and B is then consequently the value of $E_{n,n}$. The algorithm to compute the ERP distance between two time series is presented in Algorithm 2.

4.1.5 Move-split-merge

The move-split-merge (MSM), introduced by Stefan et al. in 2012, is closely related to the ERP distance function. Both of these elastic distances aim to transform one time series into another time series. The metric uses three operations: Move, Split and Merge, each of which has an associated cost. As explained in Section 4.1.4, ERP insertions and deletions cost the absolute magnitude of the value that was inserted or deleted. In MSM, the cost for insertion and deletion depends on both it's value and the adjacent values. To accommodate this, two new operations are created, Split and Merge. The Split operation duplicates a value in a time series, while the Merge function removes a value from a time series. Furthermore, MSM also uses the substitute

operation, which is referred to as Move.

The cost to move datapoint i by a value of v is equivalent to the absolute value of v, while the cost for both split and merge operations $c = 1$.

The overall cost of transforming series A into B is calculated by adding up all the individual costs of each transformations executed. However, there exist an infinite number of ways to transform series A into B using the defined operations. Therefore, the move-split-merge distance between series A and B is defined as the minimum total cost among all possible combinations of operations.

Since the MSM distance is closely related to the ERP distance, the algorithms are similarly related. This algorithm is presented in Algorithm 3. The cost function used in this algorithm is defined as:

$$C(x, y, z, c) = \begin{cases} c \text{ if } y \leq x \leq z \text{ or } y \geq x \geq z \\ c + min(|x - y|, |x - z|) \text{ otherwise} \end{cases} \tag{4}$$

4.1.6 Time warp edit

The time warp edit (TWE) distance, originally proposed by Marteau (2008), is a measure closely related to both WDTW and ERP. The method introduces a parameter ν which controls the stiffness of warping. The stiffness is weighted in the match operation, while for deletion and insertion it is simply added to the cost. An L1-norm is used for matches, with a constant penalty λ added when sequences do not match. The λ is analog to the parameter c of MSM, and as such is set to 1, whereas the ν is related to the g parameter of WDTW and set to 0.05. The algorithm of finding the TWE distance between two time series is presented in Algorithm 4 in the Appendix.

4.2 Replication: Clustering performance methods

To assess the quality of the clusterers, six performance metrics are employed. These metrics align with those used by Holder et al. (2024), to allow for comprehensive comparison between the two studies. Clustering accuracy (CL-ACC) is the fraction of correct predictions. To determine whether a cluster prediction is correct, it is necessary to assign each cluster to the class value that best matches it. This can be done by taking the maximum accuracy from every permutation of cluster and class values S_k. The formula of the CL-ACC is presented in equation 5.

$$\text{CL-ACC}(\mathbf{y}, \hat{\mathbf{y}}) = \max_{\mathbf{s} \in \mathbf{S_k}} \frac{1}{|\mathbf{y}|} \sum_{i=1}^{|y|} \begin{cases} 1, & y_i = \mathbf{s}(\hat{y_i}) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

This is method of computing the clustering accuracy is computationally expensive. To speed up the process of calculating the clustering accuracy the Hungarian method is utilized. First, the confusion matrix is created, where each row represents instances in a predicted class, and each column represents the instances in the ground truth class. Then the linear sum assignment problem is performed to find the best possible mapping between predicted cluster labels and ground truth cluster labels (Kuhn, 1955). Finally, the CL-ACC is calculated by dividing the number of correctly assigned data points by the total number of datapoints.

Another performance measure to evaluate the clusters is the Rand index (RI), first proposed by Rand (1971). The RI measures the similarity between two label sets. In this study, the Rand index is measured between the ground truth labels and the labels produced by the clustering algorithms performed and as such can be used to compare the clustering algorithms with each other. The Rand index is then calculated by dividing the number of true positives and true negatives by the total number of predictions made.

One of the main problems with the RI is that it can give disproportionate high scores for clusterings that are dissimilar, especially when the number of clusters is large. This issue is prevented by using the adjusted Rand index (ARI), proposed by Hubert and Arabie (1985). the ARI adjusts the Rand index based on the expected scores of a purely random model.

Mutual information (MI) is a statistical method that measures the amount of information that can be obtained about one variable by observing another variable. As such it measures the agreement between the ground truth labels and the predicted labels. When comparing MI across different datasets, the scores themselves are often not interpretable and as such the normalised mutual information score (NMI) is also utilized, which scales the values of the MI between 0 and 1. The mutual information score also has the same problem of the RI in which it is inflated for datasets with high number of cluster. As such, the adjusted mutual information score (AMI) is used to adjust the MI by taking into account the expected value of the MI under a null reference.

In this study, the evaluation measures will be compared across multiple datasets. To do so, the rank ordering method is utilized, explained by Demšar (2006). In his work, the post hoc Neymeni test is consulted to identify significant differences between metrics. However, by recommendation of Benavoli et al. (2016), this test is replaced by the pairwise Wilcoxon sign rank test (Wilcoxon, 1992).

4.3 Extension: Model selection

A fundamental problem in cluster analysis is how to determine the optimal number of clusters in a dataset. The terms "model selection" and "determining the number of clusters" are used interchangeably in this section. The problem of model selection has big effects in the clustering performance and is highly susceptible to changes in dataset complexity (Milligan and Cooper, 1985). As such, it is important to choose both an appropriate clustering method, as well as a method for determining the number of clusters in a dataset for accurate results. In their research, Holder et al. (2024) completely disregard this important characteristic in clustering analysis, presuming the number of clusters to be known a priori. However, in a real-world setting, the number of clusters is generally not known, underscoring the importance of this analysis for TSCL literature.

For model selection, a common method is to start with a predefined set of k values and consequently selecting the optimal k using some form of statistical measure. The problem of this approach is that it is computationally demanding, especially for problems with large number of clusters. To account for this the range of values for k is set close to the actual value. As a result, by limiting the range of k values, the number of times the clustering algorithms need to be executed is reduced.

According to the findings of Holder et al. (2024), the k-medoids clusterer performs signi-

ficantly better than k-means across all elastic distance measures investigated and as such is adopted for the purpose of model selection.

4.3.1 Silhouette index

The concept of using silhouettes to determine the optimal number of clusters was first proposed in 1987 by Rousseeuw. This approach utilizes the silhouette statistic, which is a measure of how similar an object is to its own cluster compared to other clusters. The silhouette method a quantitative method that always returns a value for the number of clusters. According to a 2013 study by Arbelaitz et al., the silhouette method performs best among 30 cluster validity indices. However, the authors also note that the performance of this method is highly susceptible to the data used.

The silhouette coefficient for observation i is computed as $s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$, where a_i is the mean distance between observation i and all other observations in its own cluster. b_i is the mean distance between observation i and all other observations in the next-nearest cluster.

the silhouette index is determined for each value of k in the range of values. This index is the average of the silhouette coefficients for all individual observations. High silhouette scores indicate that the observations from a cluster are far away from observations outside of their cluster. The value of k that yields the highest silhouette index is consequently chosen as the optimal cluster count.

4.3.2 Davies-Bouldin index

The Davies-Bouldin (DB) index is an unsupervised method for evaluating cluster methods (Davies and Bouldin, 1979). It gives higher scores to clustering where there is good separation between the clusters. Holder et al. (2024) use the index for the study on tuning parameters, however the DB index can also be used to determine the number of clusters if this is not pre-determined.

The DB score calculates a similarity measure between each cluster pairs, which is based on the distance between their centroids and their sizes. The equations for computing the DB score is presented in Appendix C Suppose an experiment is run to determine the optimal number of clusters for the distance function d.

Firstly, the centroids $\mathbf{c}_i$ are calculated for all clusters using equation 6, and A_{ij} denotes that time series j is clustered in cluster i. Thereafter, the dispersion of all the clusters is calculated using the formula in equation 7, where S_i is the dispersion of cluster i and d denotes the (elastic) distance function utilized. The distance between centroids i and j is defined as M_{ij}, given by equation 8. The similarity of cluster i is defines as R_i in equation 9. Finally, the Davies-Bouldin index is calculated as the average all similarity scores, as in equation 10.

The DB score is a measure of the similarity between clusters. If data is clustered accordingly it should have low similarity between the clusters and as such, the value of k which leads to the lowest Davies-Bouldin score is chosen as the optimal number of clusters.

4.3.3 Calinksi-Harabasz index

The Calinski-Harabasz (CH) index is a metric which denotes the ratio of the between-cluster separation to the within-cluster dispersion (Caliński and Harabasz, 1974). Higher CH scores

indicate that a clustering solution is well-seperated and compact. The Calinski-Harabasz score is defined as $CH = \frac{B_k/(k-1)}{W_k/(n-k)}$, where B_k is the between-cluster variance. It is defined as the sum of the squared distances between the centroids of each cluster and the overall centroid of the data. W_k is the within-cluster variance, defined as the sum of squared distances between each data point and centroid of its corresponding cluster.

A high CH score implies that the clusters are distinct and well-defined, making it easier to differentiate between them. As such, the value of k that lead to the highest CH score is chosen as the optimal number of clusters.

4.4 Extension: Evaluating the model selection

Despite the vast literature on how to determine the optimal number of clusters within a dataset, the evaluation measures of such methods are scarce. Often these techniques are applied to data that has not previously been clustered and are mostly evaluated using simulation studies. In contrast, this study employs these methods with pre-classified data from the UCR archive.

Raihan (2023) focuses on determining the number of clusters in both raw time series data and using data with extracted features from the UCR archive. He considered the "true" cluster count to be the the number of clusters of the UCR archive. The evaluation was based on the frequency of correctly identifying the number of clusters, how many times the predicted results were close (i.e. one more or less than the actual number) and how many times they were wrong.

In this research, the same metric is utilized. However, this study addresses a limitation in his approach, which only considered datasets where the "actual" number of clusters is less than or equal to seven. As in this research also datasets with more than seven clusters are considered, a prediction is considered "close" when the prediction is either one off the actual number, or if it is within 20% of the actual value. This revised approach allows for a more robust evaluation of the model selection methods.

5 Results

In this section the results on the time series clustering problems are presented and discussed. First, in Sections 5.1 and 5.2, the results for the replication of the study by Holder et al. (2024) are given. Sections 5.3 and 5.4 present the extension on model selection methods, where the cluster count k is not known prior to clustering. All code used can be found on the associated GitHub repository[3].

5.1 Replication: k-means clustering

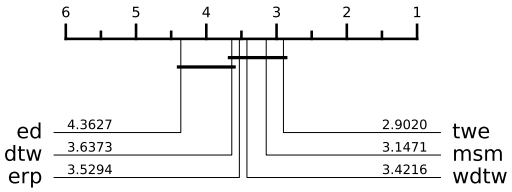
The first analysis involves the comparison of the six distance functions using k-means clustering. k-means is a widely used clustering algorithm that aims to partition a set of n observations into k clusters. Each observation is assigned to the cluster with the nearest mean. For each of the 51 UCR datasets the number of clusters k to be formed was set to "true" number of clusters for each individual dataset, as defined in the UCR archive.

[3] github.com/luckyluc99/TSCL-model-selection

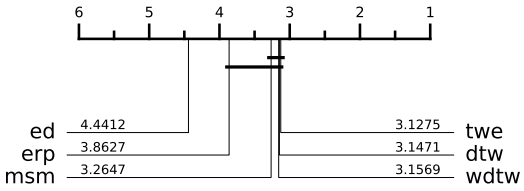
Holder et al. (2024) make an analysis on three initialization methods: random, Forgy and $k$means++. They note that the all three methods perform about as well as each other and that the difference in performance was generally not caused by the initialisation algorithm. Consequently, random initialization is chosen, as this is the fastest of the three methods. Based on the recommendation of Holder et al. (2024), the remaining parameters are set to the default settings of the aeon package[4] (maximum 300 iterations and 10 restarts).

In time series clustering an important decision is whether to perform the analysis on normalised data or on raw data. Holder et al. (2024) treat normalisation as a parameter and perform their analysis on both normalised data and on the raw data. They conclude that the same performance patterns are found, irrelevant of normalisation. As such, for this research, raw data is used.

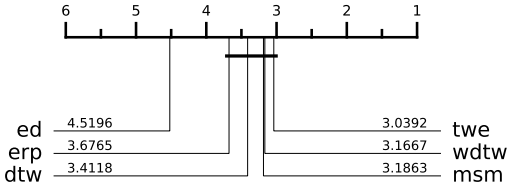
Another question of high debate in TSCL literature is whether to merge the test and training sets. For both the analysis on k-means and k-medoids clustering, all the training is done on the default training sets given by the UCR archive. This is to allow for an accurate replication of the paper by Holder et al. (2024). The results are given on both the training and test sets.



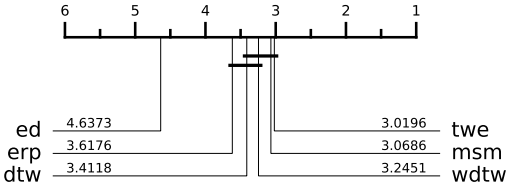
(a) Accuracy



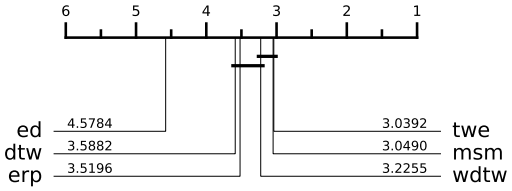
(b) Rand Index



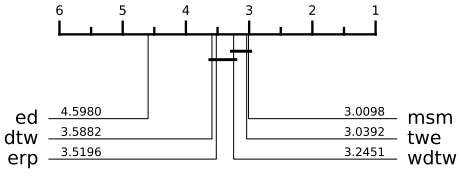
(c) Adjusted RI



(d) Mutual information



(e) Adjusted MI



(f) Normalised MI

Figure 3: Critical difference diagrams for k-means clustering using six distance functions. The analysis involves 51 UCR datasets (raw data), evaluated using six performance measures on the test data. A lower score means a distance function performs better. Distance functions grouped by horizontal lines means that there is no pairwise significant difference between them.

[4]www.aeon-toolkit.org/en/stable/api_reference/clustering.html/

The assessment of the various distance function is done using the six metrics explained in Section 4.2. A summary of these metrics for the k-means clustering using six distance functions is presented for the test and training sets in Figures 3 and 6 respectively. This critical difference diagram displays the distance functions ordered by the average rank of the metric across all datasets. Therefore, when a distance measure is further to the right in a diagram, it means it performs better. The horizontal lines denote the groups of distance functions between which there is no significant difference between them. For example, in Figure 3a, ERP has the lowest average rank of 2.8039, meaning that it performs best. However, ERP forms a group with TWE and MSM with no pairwise significant difference between them. These results are based on the test data of the UCR archive. The results on the training data can be found in Figure 6 in Appendix D.

The results largely agree with the findings of Holder et al. (2024). Figure 3 shows that for the k-means algorithm, the TWE and MSM are the top performing distance functions, with no significant difference between them. The WDTW and ERP follow closely behind in performance.

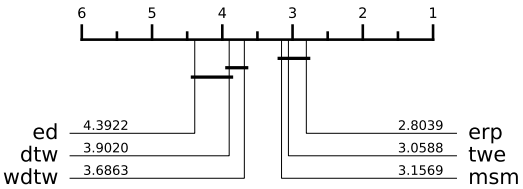
The DTW performs poorly, despite the common believe that DTW combined with k-means is the benchmark in TSCL. The Euclidean distance performs worse on all six measures, which slightly contradicts the findings by Holder et al. (2024). This discrepancy can be attributed to the different datasets used for the analysis. Holder et al. utilized 112 datasets of the UCR archive, while in this study only 51 datasets where investigated. The reason for this discrepancy can be found in Tables 7 and 9 of their report, in which they conclude that the Euclidean distance performs particularly well in datasets with large number of clusters and with large number of training cases. These are precisely the datasets that where not investigated in this study due to time constraints.

5.2 Replication: k-medoids clustering

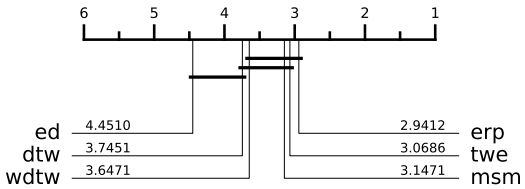
k-medoids is a clustering technique, which like k-means, aims to partition a set of n observations into k clusters. However, now each observation is assigned to the cluster that has the closest medoid. To allow for an extensive comparison between k-means and k-medoids clustering the same parameters are used (i.e. random initialization, max 300 iterations and 10 restarts).

Figure 4 presents the ranked summary statistics for the six distance functions using k-medoids clustering, on the test data. The results for the training data is presented in Figure 7 in Appendix D.

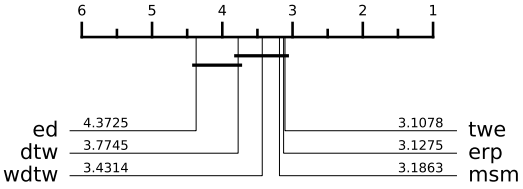
The general pattern for k-medoids clustering is the same as for the k-means algorithm. The ED continues to perform poor across all evaluated metrics, although it no longer performs significantly worse than the dynamic time warping distance. A possible explanation for the poor performance of the ED is that it performs poorly when using data that has low number of clusters, which is primarily the data used in this research. With k-medoids clustering, TWE, MSM and ERP form a top performing clique, which means there is no significant difference between the three distance measures. These top performing distance measures all apply an explicit penalty for warping when calculating the distance between two time series. Important to note that ERP ranks significantly higher with k-medoids clustering than with the k-means algorithm, as is also reported by Holder et al. (2024).



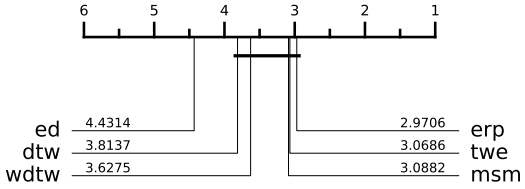
(a) Accuracy



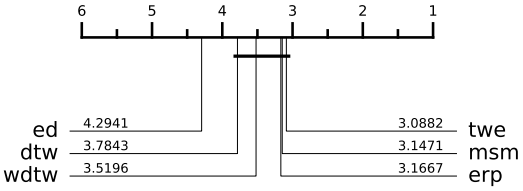
(b) Rand Index



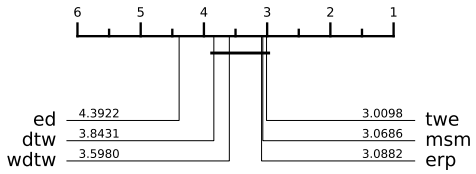
(c) Adjusted RI



(d) Mutual information



(e) Adjusted MI



(f) Normalised MI

Figure 4: Critical difference diagrams for k-medoids clustering using six distance functions. The analysis involves 51 UCR datasets (raw data), evaluated using six performance measures on the test data. A lower score means a distance function performs better. Distance functions grouped by horizontal lines means that there is no pairwise significant difference between them.

Table 3 in the appendix presents the average clustering accuracy for both k-means and k-medoids clustering. An increase in performance is realized for all distance measure except the ED. The same ordering of distances is found, with the ERP, TWE and MSM having the highest average accuracy. Notably, the significant increase in accuracy in ERP is evident, with this distance benefiting the most from the transition to the k-medoids clustering algorithm.

The average clustering accuracy of the various distance functions are substantially higher than the results reported by Holder et al. (2024). This discrepancy can be largely attributed to the different datasets employed in this research. In this study the majority of the datasets with 2 clusters are examined, and given the definition of the clustering accuracy, these dataset inherently possess a minimum accuracy of 50%. Consequently, the mean clustering accuracy is higher compared to their findings, where they utilized more datasets of large cluster count.

5.3 Extension: Model selection

This section presents the results on the model selection methods for the time series datasets of the UCR archive. This is done by first setting a range of k values and then using three criteria to determine the optimal number of clusters in this range. The lower bound of this

range was set to $k_{min} = \max(k_{true} - 10, 2)$, where k_{true} is defined as the pre-determined number of clusters for the datasets, as given by the UCR archive. The upper bound of this range was set to $k_{max} = \min(k_{true} + 10, n)$, where n denotes the number of time series in a dataset. The final constraint is placed to ensure that the clustering algorithm can execute properly, as a cluster algorithm can not partition n objects into $n + 1$ clusters, as this would result in empty clusters. The chosen range of k values was due to the time limitations of this bachelor thesis.

For the model selection analysis, the k-medoids algorithm was employed, having demonstrated superior performance in the analysis of the various distance functions. The optimal number of clusters are determined using three distinct indices: the silhouette index, Davies-Bouldin index and the Calinski-Harabasz index, as explained in Section 4.3. The scores where calculated for all values of k in the range specified and the optimal cluster count was determined according to the indices.

5.3.1 Silhouette method

The results for the silhouette method are presented in Table 1. The table is split in two, with the left side displaying the results on only the training data and the right side presenting the findings of the merged test and training sets. This combination was done to investigate whether increasing the size of the dataset could improve model selection, but also to allow for comparison with other comparative studies, like the one by Javed et al. (2020).

In Table 1, a green cell denotes a correct prediction. This occurs when the predicted number of clusters agrees with the "actual" cluster count, as provided by the UCR archive. An orange cell indicates a "close" prediction. This is the case when a prediction is either one off from the true value, or if it falls within 20% of the true value. Predictions marked with an asterisk (*) indicate that the prediction is either $k_{true} - 10$ or k_{max}.

The results for determining the number of clusters using the silhouette method show unconvincing results. When only using the training data for model selection, MSM emerges as the best performing distance measure, with it having 15 correct and 16 close predictions. This is closely followed by TWE which also has 15 correct predictions, but 15 close predictions. WDTW and ED emerge as the worse performing distance functions. When using the merged datasets, the TWE emerges as best performing distance measure, with the Euclidean distance a close second.

The increase in sample size does improve the prediction accuracy across all distance measure except for MSM. This is something that is expected as for some datasets the training set is fairly small. This phenomena can be seen in the predictions for the "MoteStrain" dataset. This dataset has an actual cluster count of 2 and the training set contains only 20 time series. When combining the training and test sets this increases to 1272 observations. When only using the training sets, only ED and MSM predict the cluster count correctly, while with the merged datasets, all distance functions have correct predictions.

The silhouette method performs well when the actual cluster count is small. However, for larger cluster counts, the method performs poorly. Especially for the datasets in group D (i.e., those with 11 or more clusters) the silhouette method fails to capture the complex clustering patterns in this group. Consequently, the method fails to produce any correct predictions for these datasets. The cause of this has to do with the definition of the silhouette index. The silhouette

index is a measure of similarity between clusters. If the number of cluster is large, the resulting clusters become more similar. The optimal number of clusters therefore get underestimated.

Table 1: Prediction of cluster count using silhouette method, for the six investigated distance function. A green cell denote a correct prediction and an orange cell denotes a "close" prediction. An asterisk (*) denotes that the prediction is either k_{min} or k_{max}. Results are on training set and with training and test set combined.

		Training						Training + test					
Data name	Actual	ED	DTW	MSM	TWE	WDTW	ERP	ED	DTW	MSM	TWE	WDTW	ERP
Adiac	37	27*	46	42	47*	28	28	32	31	44	47*	39	35
WordSynonyms	25	18	15*	16	19	24	15*	16	17	35*	27	21	15*
FacesUCR	14	10	4*	12	10	5	18	8	7	6	23	6	21
CricketZ	12	2	3	2	2	2	2	2	6	2	2	7	2
InsectWingbeatSound	11	4	2	4	2	2	3	4	2	2	4	2	2
MedicalImages	10	7	3	2	2	2	3	3	3	2	2	2	2
Fish	7	2	2	3	9	4	2	2	2	10	7	2	9
Lightning7	7	9	4	5	2	2	3	3	2	3	3	4	7
Plane	7	3	6	11	8	6	16	12	16	7	7	5	9
DistalPhalanxTW	6	2	2	2	2	2	2	2	2	2	2	2	2
MiddlePhalanxTW	6	2	2	2	2	2	2	3	2	2	2	2	2
OSULeaf	6	2	4	2	4	7	3	2	3	4	4	4	12
ProximalPhalanxTW	6	2	2	2	2	2	2	2	2	2	2	2	2
Symbols	6	7	2	6	12	11	7	5	4	4	4	6	3
SyntheticControl	6	2	2	2	2	2	3	2	4	2	2	2	3
Beef	5	7	4	4	2	7	2	4	6	6	2	3	2
ECG5000	5	2	2	2	2	3	2	2	2	2	2	3	2
DiatomSizeReduction	4	3	4	3	2	6	4	3	3	3	5	2	5
FaceFour	4	2	3	5	3	5	2	2	2	7	5	3	7
Trace	4	2	3	2	2	3	2	2	3	2	2	3	2
ArrowHead	3	2	2	2	2	10	2	3	2	2	2	2	2
BME	3	2	2	2	2	2	2	2	2	2	2	2	2
CBF	3	2	3	5	3	4	2	2	3	2	3	2	3
ChlorineConcentration	3	2	3	4	4	3	4	2	3	2	2	3	2
DistalPhalanxOutlineAgeGroup	3	2	2	2	2	2	2	2	2	2	2	2	2
Meat	3	2	2	2	2	2	2	2	2	2	2	2	2
MiddlePhalanxOutlineAgeGroup	3	2	2	2	2	2	2	3	2	2	2	2	2
ProximalPhalanxOutlineAgeGroup	3	2	2	2	2	2	2	2	2	2	2	2	2
SmoothSubspace	3	4	2	2	4	2	5	2	2	2	2	2	2
UMD	3	2	2	2	2	2	2	2	3	2	2	2	2
Chinatown	2	2	5	2	2	5	5	2	2	2	2	2	2
Coffee	2	2	2	2	2	2	2	2	3	4	2	3	4
ECG200	2	5	2	2	3	2	4	2	2	3	3	2	7
ECGFiveDays	2	3	4	3	3	7	3	2	11	4	2	6	3
FreezerRegularTrain	2	4	2	2	2	2	2	3	3	2	2	3	2
FreezerSmallTrain	2	2	3	2	2	2	2	3	3	2	2	3	2
GunPoint	2	5	3	2	2	2	4	2	2	2	2	2	2
GunPointAgeSpan	2	2	3	9	2	4	2	2	3	4	4	3	3
GunPointMaleVersusFemale	2	3	3	4	5	4	3	2	3	4	4	3	3
GunPointOldVersusYoung	2	2	3	3	4	3	2	2	3	4	4	3	3
Ham	2	2	3	2	2	2	2	4	3	2	2	3	2
ItalyPowerDemand	2	4	2	7	3	2	6	4	2	2	5	2	2
MoteStrain	2	2	3	2	3	3	3	2	2	2	2	2	2
PowerCons	2	3	5	2	2	10	2	3	7	2	2	3	2
ShapeletSim	2	2	2	11	11	12*	12*	2	2	2	6	2	2
SonyAIBORobotSurface1	2	4	2	2	2	3	2	2	3	4	2	2	2
SonyAIBORobotSurface2	2	2	4	2	2	12*	3	4	2	2	2	2	3
ToeSegmentation1	2	7	7	2	2	6	2	3	2	2	4	2	2
ToeSegmentation2	2	10	2	6	2	2	6	2	2	2	2	3	3
TwoLeadECG	2	3	2	2	2	2	2	2	5	4	2	2	2
Wine	2	2	2	2	2	2	2	2	3	2	2	3	4
Number of correct predictions:		10	12	15	15	11	12	16	12	14	17	12	14
Number of close predictions:		16	19	16	15	16	14	16	20	14	16	23	17

One could also argue about what it means for a dataset to have a large number of clusters. Clustering is used more as an exploratory tool than a predictive tool. This means that clustering is used to analyse and interpret data to discover patterns. For clustering analysis in biology for example, it does not make sense to group the patients in large number of clusters, because then the data reduction aspect of the method is lost Grant et al. (2020). Such examples also occur in other research areas where TSCL is utilized. As such, researchers often are not interested in clustering algorithms that partition data in a large number of clusters, and there is skepticism about whether the UCR archive is able to replicate real-life TSCL problems.

5.3.2 Davies-Bouldin index

Another method for determining the optimal number of clusters in a dataset is using the DB index. The results on both the training set and the combined training and test set are presented in Table 4. The DB method performs significantly worse than the silhouette method. When only using the training set, the WDTW emerges as the best performing, however still performs poorly. Combining the training and test set improves the accuracy, however it is still worse than the silhouette method.

A possible explanation is that the DB score is closely related to the k-means algorithm. k-means is designed to optimise with respect to the average distances within a cluster, which is the same statistic that is used in the DB index. k-medoids clusters are based on the pairwise distances between the observations. The use of the k-medoids clustering algorithm could be the reason for the poor performance of the DB index.

5.3.3 Calinksi-Harabasz index

The third model selection method uses the CH index, as mentioned in Section 4.3.3. The results are presented in Table 5. The CH performs good when the cluster count is small. It does, however, suffer from the same problem as the silhouette index, where it under predicts the cluster count when the "true" number of clusters in a dataset is large. The reason for this is that both of these methods are related, as they both measure an observation from its own cluster against that of other clusters.

The number of correct predictions for silhouette, DB and CH for the six distance functions is presented in Figure 8 in Appendix D. The silhouette method combined with either MSM or TWE is the best performing method. The WDTW in combination with the CH shows good performance. This method performs especially well in datasets with moderate number of clusters (i.e. datasets in group C). The performance of the DB index is consistently poor, regardless of the distance function utilized. Therefore, it is not advisable for TSCL practitioners to use this method for the purpose of model selection.

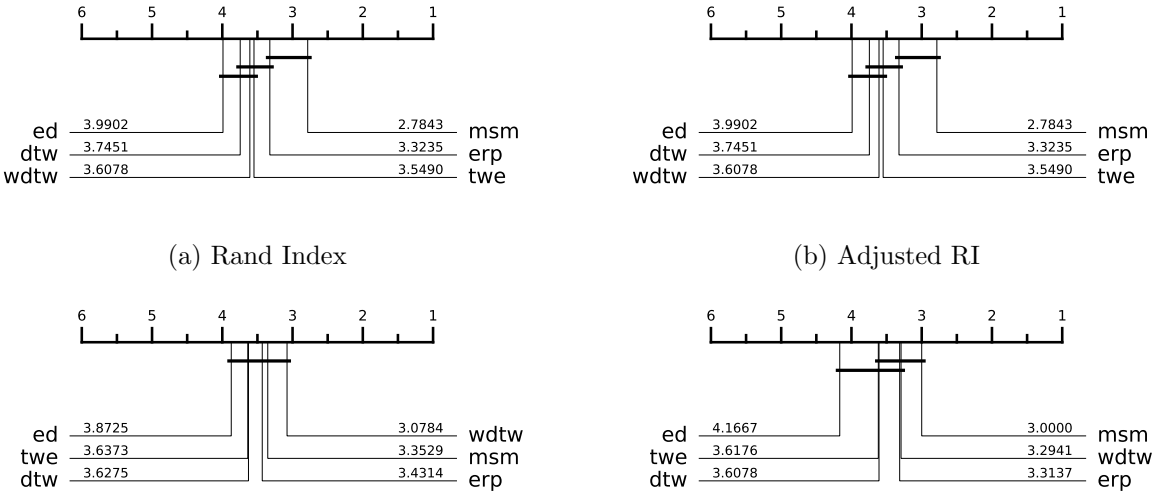
5.4 Extension: Model selection clustering performance

In Sections 5.1 and 5.2, the performance of various distance functions is evaluated with a predefined cluster count k. However, this scenario is not an accurate representation of real-world TSCL applications, where often the number of clusters is not known prior to clustering. Therefore, in Section 5.3, three methods for determining the number of clusters are presented and

discussed. This resulted in the Silhouette method being the best of the three. However, this accuracy is only with respect to how good each method was able to replicate the "real" number of clusters in the datasets of the UCR archive. An argument could be made against this evaluation method, as the number of clusters in the UCR archive might not be representative of real world TSCL applications. As such, the clusters formed by the silhouette method are evaluated against each other using the metrics explained in Section 4.2.

To do this, the k-medoids algorithm of Section 5.2 is used. As said before, this requires the number of cluster k to be pre-determined. Instead of the pre-determined cluster count of the UCR archive, now the cluster count is set to the predicted number of clusters by the Silhouette method. The training data is utilized for determining the number of clusters (i.e. the left side of Table 1), as well as for the creation of the clusters. Thereafter, the test sets are used for evaluation purposes.

The analysis is done for all of the six distance functions explained in Section 4.1. For each distance function, first the number of clusters in a dataset is predicted using the Silhouette method on the training set. Thereafter, the clusters are made using also the training set and the labels are predicted of the test sets. The methods are compared using the (adjusted) rand index and (adjusted) mutual information. This is to only compare the clusters against each other, and not to the pre-determined clusters by the UCR archive.



(a) Rand Index

(b) Adjusted RI

(c) Mutual information

(d) Adjusted MI

Figure 5: Critical difference diagrams for k-medoids clustering with prediction a full model. This involves first determining the number of clusters using the silhouette method and thereafter performing the clustering on this prediction.

Figure 5 denotes the critical difference diagrams for this analysis. The MSM and ERP emerge as best performing algorithms, with no pairwise significant difference between them. The ED and DTW perform worse among all investigated metrics. As such, the results largely resemble with the analysis in Sections 5.1 and 5.2. This ones again confirms the excellent performance of elastic distance functions in time series clustering.

6 Conclusion

In this paper, a comprehensive study is performed which compares various distance measures when used to cluster time series of the UCR archive. Six distance functions are investigated with both k-means and k-medoids clustering algorithms. This is to allow for an extensive comparison between the two algorithms, but also between the distance functions. Furthermore, an extensive analysis is given in the realm of model selection methods for TSCL.

It is concluded that the k-medoids clusterer performs better than the k-means clusterer for TSCL problems on the UCR datasets. This remark is of high importance to TSCL practitioners, as k-means clustering is used more often as a benchmark than the k-medoids clusterer. When using the k-medoids clusterer, an increase in performance is found for all distance measures, except for the Euclidean distance.

In TSCL, the best approach is to use k-medoids clustering with TWE, MSM or ERP as these elastic distance measures lead to the highest average accuracy across the investigated UCR datasets. This finding is of high importance to TSCL practitioners as the DTW with k-means is commonly used as the state-of-the-art time series clustering algorithm. In this research, the DTW only outperformed the Euclidean distance and was beaten by all other elastic distance measures. It was also concluded that k-medoids clustering outperforms k-means clustering, regardless of the elastic distance used.

A significant challenge with k-means and k-medoids clustering is that it requires the number of clusters k to be known prior to clustering. Therefore, an analysis is performed where the number of clusters k is not known prior to clustering. Three methods are considered for the model selection: silhouette index, Davies-Bouldin index and the Calinski-Harabasz index. The results are consequently evaluated against the pre-determined number of clusters as defined by the UCR datasets. The silhouette and CH method emerging as the best method, especially when the cluster count was small, however with increasing number of clusters the method performs really poorly. The Davies-Bouldin method performs worse and as such should not be used to determine the number of clusters in time series clustering.

The evaluation of the model selection is predicated on the assumption that the number of clusters in the UCR archive is the correct number of clusters. However, doubts exist regarding the UCR archive's ability to accurately represent real-world TSCL problems. When assessing the individual cluster performance, independent of the UCR archive, using the RI and MI, the MSM and ERP emerge as the most effective elastic distance measures. This ones again confirms the previously found performance of elastic distance measures in time series clustering.

In light of the findings presented in this study, several avenues for future research emerge to enhance the understanding and applicability of time series clustering. Determining the number of clusters using raw data did not show promising results, especially for large cluster counts. One promising direction for further research is to expand the model selection methods to include feature extraction methods. Prior research showed good performance of a Bag of Words vector approach and a TF-IDF vectors approach. Both of these methods are popular techniques in Natural Language Processing studies. By first performing model selection using feature extraction and thereafter using elastic distance functions to perform the clustering, one could possibly find a new state-of-the-art time series clustering model.

References

Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review. *Information systems*, 53:16–38, 2015.

Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M Pérez, and Iñigo Perona. An extensive comparative study of cluster validity indices. *Pattern recognition*, 46(1):243–256, 2013.

Alessio Benavoli, Giorgio Corani, and Francesca Mangili. Should we really use post-hoc tests based on mean-ranks? *The Journal of Machine Learning Research*, 17(1):152–161, 2016.

Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pages 359–370, 1994.

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

Lei Chen and Raymond Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803, 2004.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, pages 224–227, 1979.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. *ACM Sigmod Record*, 23(2):419–429, 1994.

Richard W Grant, Jodi McCloskey, Meghan Hatfield, Connie Uratsu, James D Ralston, Elizabeth Bayliss, and Chris J Kennedy. Use of latent class analysis and k-means clustering to identify complex patient profiles. *JAMA network open*, 3(12):e2029068–e2029068, 2020.

Christopher Holder, Matthew Middlehurst, and Anthony Bagnall. A review and evaluation of elastic distance functions for time series clustering. *Knowledge and Information Systems*, 66 (2):765–809, 2024.

Bing Hu, Yanping Chen, and Eamonn Keogh. Classification of streaming time series under more realistic assumptions. *Data mining and knowledge discovery*, 30(2):403–437, 2016.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

F. Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):67–72, 1975. doi: 10.1109/TASSP.1975.1162641.

Ali Javed, Byung Suk Lee, and Donna M Rizzo. A benchmark study on time series clustering. *Machine Learning with Applications*, 1:100001, 2020.

Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. *Pattern recognition*, 44(9):2231–2240, 2011.

R Kanakarathinam. International journal of advance research in computer science and management studies. *International Journal*, 5(1), 2017.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Baptiste Lafabregue, Jonathan Weber, Pierre Gançarski, and Germain Forestier. End-to-end deep representation learning for time series clustering: a comparative study. *Data mining and knowledge discovery*, 36(1):29–81, 2022.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28 (2):129–137, 1982.

Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. Learning representations for time series clustering. *Advances in neural information processing systems*, 32, 2019.

Pierre-François Marteau. Time warp edit distance with stiffness adjustment for time series matching. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):306–318, 2008.

Glenn W Milligan and Martha C Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1855–1870, 2015.

Md Nishat Raihan. Determining the optimal number of clusters for time series datasets with symbolic pattern forest. *arXiv preprint arXiv:2310.00820*, 2023.

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

Teemu Räsänen and Mikko Kolehmainen. Feature-based clustering for electricity use time series data. In *Adaptive and Natural Computing Algorithms: 9th International Conference, ICANNGA 2009, Kuopio, Finland, April 23-25, 2009, Revised Selected Papers 9*, pages 401–412. Springer, 2009.

Chotirat Ann Ratanamahatana and Eamonn Keogh. Everything you know about dynamic time warping is wrong. In *Third workshop on mining temporal and sequential data*, volume 32. Citeseer, 2004.

LKPJ Rdusseeun and P Kaufman. Clustering by means of medoids. In *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*, volume 31, 1987.

Lior Rokach and Oded Maimon. Clustering methods. *Data mining and knowledge discovery handbook*, pages 321–352, 2005.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. doi: 10.1109/TASSP.1978.1163055.

Alexandra Stefan, Vassilis Athitsos, and Gautam Das. The move-split-merge metric for time series. *IEEE transactions on Knowledge and Data Engineering*, 25(6):1425–1438, 2012.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics: Methodology and distribution*, pages 196–202. Springer, 1992.

Hui Zhang, Tu Bao Ho, Yang Zhang, and M-S Lin. Unsupervised feature extraction for time series clustering using orthogonal wavelet transform. *Informatica*, 30(3), 2006.

A Appendix: Data names

Table 2: Datasets used for the final analysis, together with ground-truth number of clusters k.

Data Name	k	Data Name	k
Adiac	37	MiddlePhalanxOutlineAgeGroup	3
WordSynonyms	25	ProximalPhalanxOutlineAgeGroup	3
FacesUCR	14	SmoothSubspace	3
CricketZ	12	UMD	3
InsectWingbeatSound	11	Chinatown	2
MedicalImages	10	Coffee	2
Fish	7	ECG200	2
Lightning7	7	ECGFiveDays	2
Plane	7	FreezerRegularTrain	2
DistalPhalanxTW	6	FreezerSmallTrain	2
MiddlePhalanxTW	6	GunPoint	2
OSULeaf	6	GunPointAgeSpan	2
ProximalPhalanxTW	6	GunPointMaleVersusFemale	2
Symbols	6	GunPointOldVersusYoung	2
SyntheticControl	6	Ham	2
Beef	5	ItalyPowerDemand	2
ECG5000	5	MoteStrain	2
DiatomSizeReduction	4	PowerCons	2
FaceFour	4	ShapeletSim	2
Trace	4	SonyAIBORobotSurface1	2
ArrowHead	3	SonyAIBORobotSurface2	2
BME	3	ToeSegmentation1	2
CBF	3	ToeSegmentation2	2
ChlorineConcentration	3	TwoLeadECG	2
DistalPhalanxOutlineAgeGroup	3	Wine	2
Meat	3		

B Appendix: Algorithms

Algorithm 1 DTW algorithm with Sakoe-Chiba bands (A, B (both series of length m), w_{sc} (window proportion, default $w_{sc} \leftarrow 0.2$), M (pointwise distance matrix)

Let C be an $(m+1) \times (m+1)$ matrix initialized to zero, indexed from zero.
for $i \leftarrow 1$ to m **do**
    for $j \leftarrow 1$ to m **do**
        if $|i - j| < w_{sc} \cdot m$ **then**
            $C_{i,j} \leftarrow M_{i,j} + \min(C_{i-1,j-1}, C_{i-1,j}, C_{i,j-1})$
        end if
    end for
end for
return $C_{m,m}$

Algorithm 2 ERP algorithm (A, B (both series of length m), g, (penalty value, default $g \leftarrow 0.05$))

Let E be an $(m+1) \times (m+1)$ matrix initialised to zero, indexed from zero.
for $i \leftarrow 1$ to m **do**
    for $j \leftarrow 1$ to m **do**
        if $i = 0$ **then**
            $E_{i,j} \leftarrow \sum_{k=1}^{m} |b_k - g|$
        else if $j = 0$ **then**
            $E_{i,j} \leftarrow \sum_{k=1}^{m} |a_k - g|$
        else
            $match \leftarrow E_{i-1,j-1} + |a_i - b_j|$
            $insert \leftarrow E_{i-1,j} + |a_i - g|$
            $delete \leftarrow E_{i,j-1} + |b_j - g|$
            $E_{i,j} \leftarrow \min(match, insert, delete)$
        end if
    end for
end for
return $E_{m,m}$

Algorithm 3 MSM algorithm (A, B (both series of length m), c (minimum cost, default $c \leftarrow 1$), d (pointwise distance function), $\mathbf{C}$ (cost function))

Let D be an $m \times m$ matrix initialised to zero.
$D_{1,1} = d(a_1, b_1)$
for $i \leftarrow 2$ to m **do**
    $D_{i,1} = D_{i-1,1} + \mathbf{C}(a_i, a_{i-1}, b_1, c)$
end for
for $i \leftarrow 2$ to m **do**
    $D_{1,i} = D_{1,i-1} + \mathbf{C}(b_i, a_1, b_{i-1}, c)$
end for
for $i \leftarrow 2$ to m **do**
    for $j \leftarrow 2$ to m **do**
        $match \leftarrow D_{i-1,j-1} + |a_i - b_j|$
        $insert \leftarrow D_{i-1,j} + \mathbf{C}(a_i, a_{i-1}, b_j, c)$
        $delete \leftarrow D_{i,j-1} + \mathbf{C}(b_j, b_{j-1}, a_i, c)$
        $D_{i,j} \leftarrow \min(match, insert, delete)$
    end for
end for
return $D_{m,m}$

Algorithm 4 TWE algorithm (A, B (both series of length m), λ (edit cost, default $\lambda \leftarrow 0.05$), ν (warping penalty factor, default $\nu \leftarrow 1$))

Let D be an $(m+1) \times (n+1)$ matrix initialised to 0
$D_{0,0} = 0$
for $i \leftarrow 1$ to m **do**
    $D_{i,0} = \infty$
end for
for $i \leftarrow 1$ to n **do**
    $D_{0,i} = \infty$
end for
for $i \leftarrow 1$ to m **do**
    for $j \leftarrow 1$ to n **do**
        $match \leftarrow D_{i-1,j-1} + |a_i - b_j| + |a_{i-1} - b_{j-1}| + 2\nu|i - j|$
        $delete \leftarrow D_{i-1,j} + |a_i - a_{i-1}| + \lambda + \nu$
        $insert \leftarrow D_{i,j-1} + |b_j - b_{j-1}| + \lambda + \nu$
        $D_{i,j} \leftarrow \min(match, insert, delete)$
    end for
end for
return $D(m, n)$

C Appendix: Davies-Bouldin equations

$$\mathbf{c}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} A_{ij} \tag{6}$$

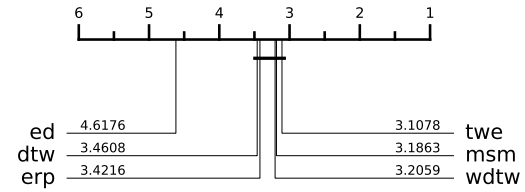
$$S_i = \frac{1}{n_i} \sum_{j=1}^{n_i} d(A_{ij}, \mathbf{c}_i) \tag{7}$$

$$M_{ij} = d(\mathbf{c}_i, \mathbf{c}_j) \tag{8}$$

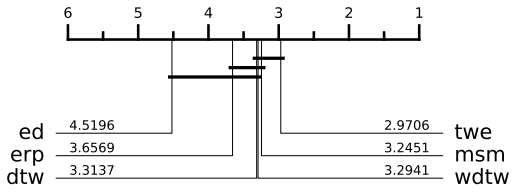
$$R_i = \max_{i \neq j} \left(\frac{S_i + S_j}{M_{ij}} \right) \tag{9}$$

$$\mathrm{DB} = \frac{1}{k} \sum_{i=1}^{k} R_i \tag{10}$$

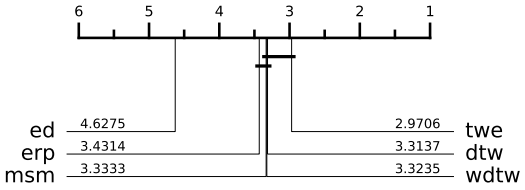
D Appendix: Additional results



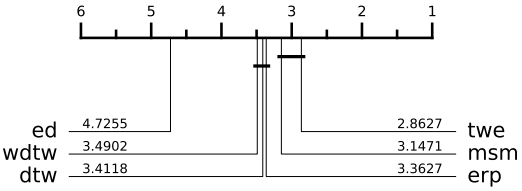
(a) Accuracy



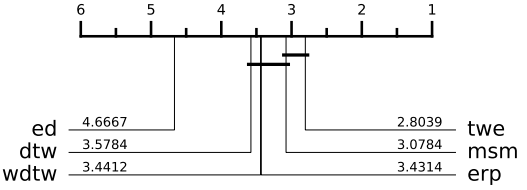
(b) Rand Index



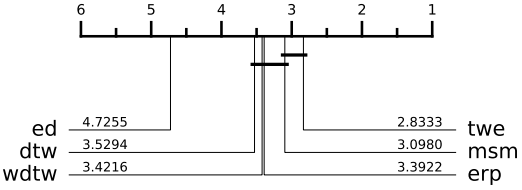
(c) Adjusted RI



(d) Mutual information

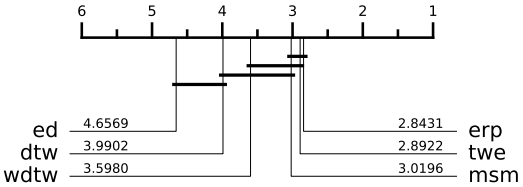


(e) Adjusted MI

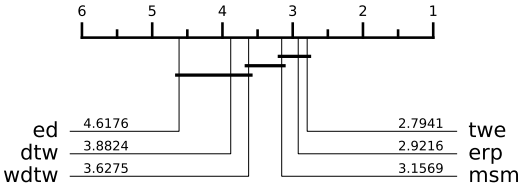


(f) Normalised MI

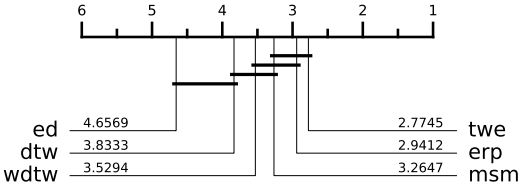
Figure 6: Critical difference diagrams for k-means clustering using six distance functions. The analysis involves 51 UCR datasets (raw data), evaluated using six performance measures on the train data. A lower score means a distance function performs better. Distance functions grouped by horizontal lines means that there is no pairwise significant difference between them.



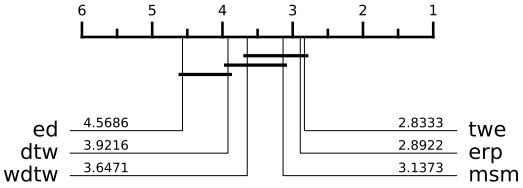
(a) Accuracy



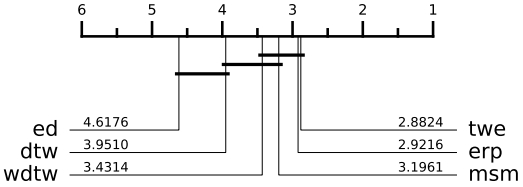
(b) Rand Index



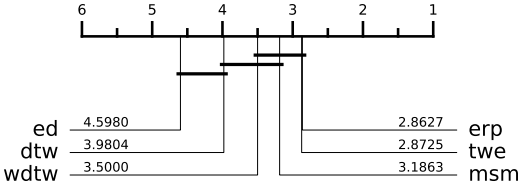
(c) Adjusted RI



(d) Mutual information



(e) Adjusted MI



(f) Normalised MI

Figure 7: Critical difference diagrams for k-medoids clustering using six distance functions. The analysis involves 51 UCR datasets (raw data), evaluated using six performance measures on the train data. A lower score means a distance function performs better. Distance functions grouped by horizontal lines means that there is no pairwise significant difference between them.

Table 3: Average clustering accuracy across the 51 UCR datasets for both k-means and k-medoids clustering. Results are on the test dataset.

Distance	k-means (%)	k-medoids (%)	Difference (%)
ERP	60.87	63.69	2.82
TWE	62.33	63.22	0.88
MSM	62.43	62.79	0.36
WDTW	60.33	60.69	0.36
DTW	60.24	60.45	0.21
ED	57.34	57.11	-0.23

Table 4: Prediction of the cluster count using Davies-Bouldin score, for the six investigated distance function. A green cell denote a correct prediction and an orange cell denotes a "close" prediction. An asterisk (*) denotes that the prediction is either k_{min} or k_{max}. Results are on train set and with training and test set combined.

		Train						Train + test					
Data name	Actual	ED	DTW	MSM	TWE	WDTW	ERP	ED	DTW	MSM	TWE	WDTW	ERP
Adiac	37	43	33	41	46	44	47*	27*	27*	32	37	31	31
WordSynonyms	25	35*	29	35*	34	30	31	25	20	30	34	34	34
FacesUCR	14	24*	17	24*	21	24*	23	4*	4*	21	24*	24*	16
CricketZ	12	2	3	2	2	2	3	2	22*	7	2	3	2
InsectWingbeatSound	11	5	2	7	3	2	20	2	2	2	4	2	2
MedicalImages	10	4	3	8	2	3	4	3	3	2	3	2	3
Fish	7	2	2	15	17*	16	15	12	2	17*	8	17*	9
Lightning7	7	17*	17*	17*	16	17*	17*	17*	17*	9	3	5	4
Plane	7	4	6	6	8	6	17*	4	4	5	7	5	9
DistalPhalanxTW	6	2	2	2	2	2	2	2	2	2	2	2	2
MiddlePhalanxTW	6	2	2	2	2	2	2	2	2	2	2	2	2
OSULeaf	6	2	16*	16*	2	5	16*	3	3	2	12	3	16*
ProximalPhalanxTW	6	2	2	2	2	2	2	2	2	2	2	2	2
Symbols	6	16*	16*	16*	16*	14	15	4	3	4	4	2	5
SyntheticControl	6	2	2	2	2	2	2	2	2	2	2	2	3
Beef	5	13	13	15*	15*	4	13	8	15*	14	14	3	13
ECG5000	5	2	2	2	2	8	2	2	2	2	2	6	2
DiatomSizeReduction	4	14*	14*	13	13	14*	14*	4	4	5	5	5	3
FaceFour	4	14*	14*	12	14*	14*	12	14*	14*	9	5	14*	14*
Trace	4	5	2	2	2	2	2	2	2	2	4	2	3
ArrowHead	3	13*	2	9	11	10	10	2	3	2	2	3	2
BME	3	2	2	2	13*	2	2	2	2	2	2	2	2
CBF	3	13*	12	13*	12	11	13*	7	2	2	2	2	4
ChlorineConcentration	3	4	4	4	4	4	4	4	4	4	3	3	4
DistalPhalanxOutlineAgeGroup	3	2	2	2	2	2	2	2	2	2	2	2	2
Meat	3	2	2	2	2	2	2	2	2	2	2	2	2
MiddlePhalanxOutlineAgeGroup	3	2	2	2	2	2	2	2	2	2	2	2	2
ProximalPhalanxOutlineAgeGroup	3	2	2	2	2	2	2	2	2	2	2	2	2
SmoothSubspace	3	13*	7	13*	9	13*	5	2	2	2	2	2	2
UMD	3	2	2	2	2	2	2	2	2	2	2	2	2
Chinatown	2	12*	8	11	12*	8	12*	2	2	2	2	2	2
Coffee	2	12*	7	12*	12*	12*	12*	2	2	6	5	3	6
ECG200	2	5	2	2	11	2	4	2	2	6	5	5	5
ECGFiveDays	2	12*	10	12*	12*	10	12*	8	5	4	4	6	3
FreezerRegularTrain	2	11	3	7	6	2	6	3	3	3	3	3	3
FreezerSmallTrain	2	12*	2	9	9	7	9	3	3	3	3	3	3
GunPoint	2	5	3	7	4	2	5	2	2	2	2	2	2
GunPointAgeSpan	2	2	3	2	2	2	2	2	3	2	2	3	3
GunPointMaleVersusFemale	2	3	3	3	5	2	3	2	3	2	2	3	3
GunPointOldVersusYoung	2	2	3	3	2	3	2	2	3	2	2	3	3
Ham	2	12*	11	10	12*	12*	12*	8	9	11	11	11	12*
ItalyPowerDemand	2	20	18	4	3	2	3	7	6	2	5	2	3
MoteStrain	2	12*	12*	12*	12*	12*	10	2	2	2	2	2	2
PowerCons	2	3	5	2	2	5	2	2	9	2	2	3	2
ShapeletSim	2	12*	12*	12*	12*	12*	12*	12*	12*	12*	12*	12*	12*
SonyAIBORobotSurface1	2	12*	11	11	9	12*	12*	7	5	6	5	4	2
SonyAIBORobotSurface2	2	12*	12*	12*	12*	12*	12*	3	3	2	2	11	7
ToeSegmentation1	2	12*	11	12*	12*	10	12*	4	4	3	4	2	2
ToeSegmentation2	2	12*	12*	12*	12*	12*	12*	12*	5	2	8	3	12*
TwoLeadECG	2	11	2	12*	9	11	2	2	11	5	2	2	2
Wine	2	7	2	10	10	2	10	4	4	11	2	4	4
Number of correct predictions:		2	4	3	3	7	4	12	7	10	14	8	7
Number of close predictions:		11	16	12	8	13	9	12	16	16	14	19	22

Table 5: Prediction of the cluster count using Calinski-Harabasz score, for the six investigated distance function. A green cell denote a correct prediction and an orange cell denotes a "close" prediction. An asterisk (*) denotes that the prediction is either k_{min} or k_{max}. Results are on train set and with training and test set combined.

		Train						Train + test					
Data name	Actual	ED	DTW	MSM	TWE	WDTW	ERP	ED	DTW	MSM	TWE	WDTW	ERP
Adiac	37	27*	27*	41	27*	28	28	28	31	32	36	39	29
WordSynonyms	25	15*	16	18	19	16	15*	16	15*	17	25	21	15*
FacesUCR	14	4*	5	9	6	5	6	4*	4*	6	4*	5	6
CricketZ	12	2*	3	2*	2*	2*	2*	2*	6	2*	2*	3	2*
InsectWingbeatSound	11	4	8	7	3	2	2	6	4	2	4	2	3
MedicalImages	10	4	3	2	3	3	2	3	3	4	3	5	3
Fish	7	2	2	7	8	15	5	2	2	5	7	6	7
Lightning7	7	6	4	5	4	2	2	3	2	2	2	2	4
Plane	7	3	6	2	9	6	3	7	7	7	4	5	7
DistalPhalanxTW	6	2	2	2	2	2	2	2	2	2	2	2	2
MiddlePhalanxTW	6	3	2	2	2	2	2	3	2	3	3	2	3
OSULeaf	6	2	4	2	2	6	3	2	3	4	6	3	9
ProximalPhalanxTW	6	2	2	2	2	2	2	2	2	2	2	2	2
Symbols	6	6	2	6	8	10	4	5	3	3	4	6	5
SyntheticControl	6	2	2	2	2	2	2	2	2	2	2	2	2
Beef	5	5	4	4	4	4	4	4	6	6	5	3	7
ECG5000	5	2	2	2	2	3	2	2	2	2	2	3	2
DiatomSizeReduction	4	3	3	5	6	5	3	3	4	3	3	3	3
FaceFour	4	2	8	5	3	5	2	2	2	2	5	3	3
Trace	4	2	2	2	2	2	2	2	2	2	2	2	2
ArrowHead	3	2	2	2	2	5	2	3	2	2	2	2	2
BME	3	2	2	2	2	2	2	2	2	2	2	2	2
CBF	3	2	4	3	3	3	3	2	3	2	2	2	2
ChlorineConcentration	3	2	4	4	2	4	4	2	2	2	2	2	2
DistalPhalanxOutlineAgeGroup	3	2	2	2	2	2	2	2	2	2	2	2	2
Meat	3	2	2	2	2	2	2	2	2	2	2	2	2
MiddlePhalanxOutlineAgeGroup	3	2	2	2	2	2	2	3	2	3	3	2	3
ProximalPhalanxOutlineAgeGroup	3	2	2	2	2	2	2	2	2	2	2	2	2
SmoothSubspace	3	3	6	2	4	6	5	2	2	2	2	2	2
UMD	3	2	2	2	2	2	2	2	2	2	2	2	2
Chinatown	2	2	4	3	2	4	4	2	2	4	2	2	2
Coffee	2	2	2	2	2	2	2	2	2	4	2	3	4
ECG200	2	5	6	2	3	2	4	2	2	3	3	2	5
ECGFiveDays	2	2	10	4	2	2	3	3	4	11	2	4	3
FreezerRegularTrain	2	4	3	7	4	2	6	3	3	3	3	3	3
FreezerSmallTrain	2	2	3	3	2	2	3	3	3	3	3	3	3
GunPoint	2	5	3	6	2	2	4	2	2	2	2	2	2
GunPointAgeSpan	2	3	3	5	2	5	3	3	3	4	4	3	3
GunPointMaleVersusFemale	2	3	3	5	4	5	3	3	3	4	4	3	3
GunPointOldVersusYoung	2	3	3	3	4	3	3	3	3	4	4	3	3
Ham	2	3	3	2	2	2	3	4	3	2	3	3	2
ItalyPowerDemand	2	4	4	3	4	4	3	4	2	2	5	2	3
MoteStrain	2	2	3	2	3	2	3	2	2	2	2	2	2
PowerCons	2	3	5	2	2	5	2	3	7	2	2	8	2
ShapeletSim	2	2	2	11	8	6	11	3	4	7	8	2	12*
SonyAIBORobotSurface1	2	4	5	2	4	3	2	2	3	4	6	2	2
SonyAIBORobotSurface2	2	2	5	2	2	5	3	3	3	2	2	5	7
ToeSegmentation1	2	3	4	3	3	2	3	3	3	2	2	2	2
ToeSegmentation2	2	10	4	6	2	2	2	2	3	3	2	2	2
TwoLeadECG	2	3	8	9	3	7	2	2	11	2	2	2	2
Wine	2	2	2	2	2	2	2	2	3	11	3	3	3
Number of correct predictions:		11	3	11	12	13	7	12	9	10	15	11	12
Number of close predictions:		18	20	18	16	13	20	21	22	16	17	23	20

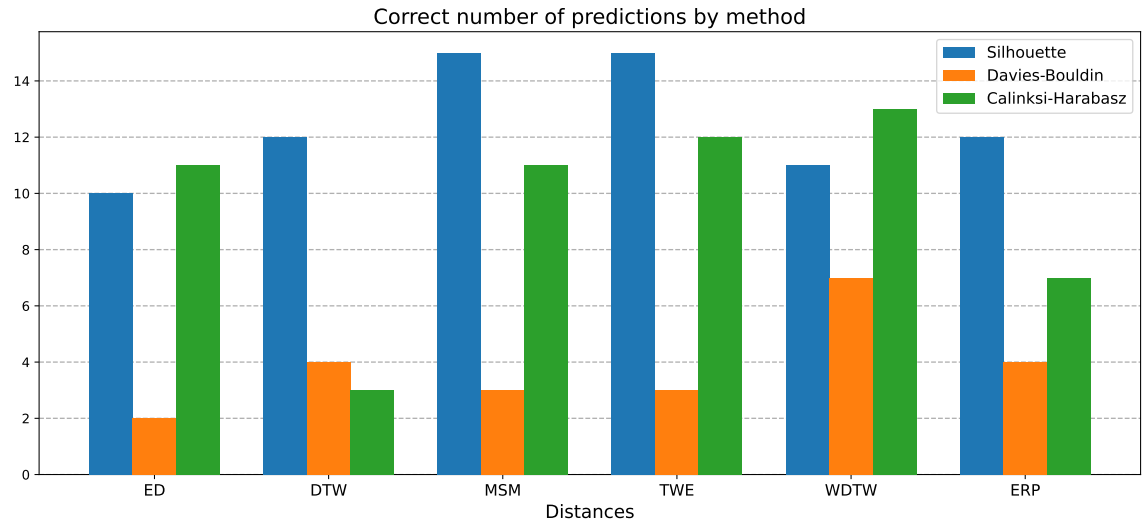
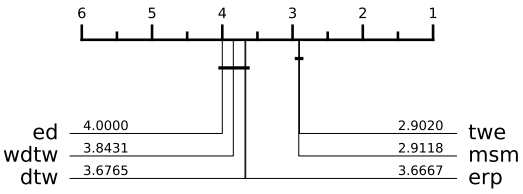
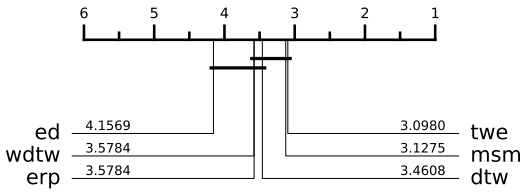


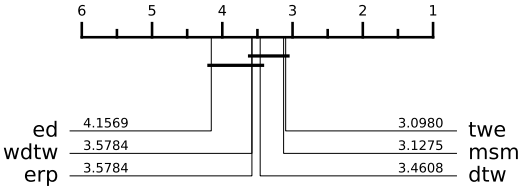
Figure 8: Comparison of the correct prediction between the three methods.



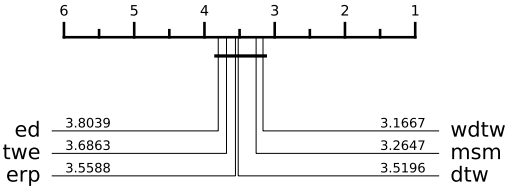
(a) Accuracy



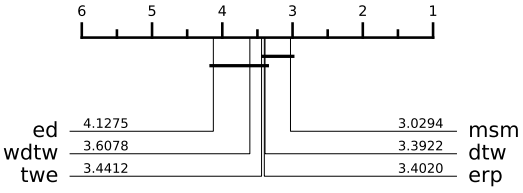
(b) Rand Index



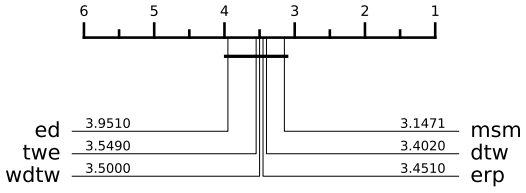
(c) Adjusted RI



(d) Mutual information



(e) Adjusted MI



(f) Normalised MI

Figure 9: Critical difference diagrams for k-medoids clustering with prediction a full model. This involves first determining the number of clusters using the silhouette method and thereafter performing the clustering on this prediction. A lower score means a distance function performs better. Distance functions grouped by horizontal lines means that there is no pairwise significant difference between them. Results are on the train set.