Machine Learning with Applications 1 (2020) 100001

Contents lists available at ScienceDirect

Machine Learning with Applications

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

A benchmark study on time series clustering

Ali Javed [a,c,*], Byung Suk Lee [a,c], Donna M. Rizzo [b,c]

[a] Department of Computer Science, University of Vermont, Burlington, VT, USA
[b] Department of Civil and Environmental Engineering, University of Vermont, Burlington, VT, USA
[c] Gund Institute for Environment, University of Vermont, Burlington, VT, USA

ARTICLE INFO

Keywords:
Time series
Clustering
Benchmark
UCR archive

ABSTRACT

This paper presents the first time series clustering benchmark utilizing all time series datasets currently available in the University of California Riverside (UCR) archive — the state of the art repository of time series data. Specifically, the benchmark examines eight popular clustering methods representing three categories of clustering algorithms (partitional, hierarchical and density-based) and three types of distance measures (Euclidean, dynamic time warping, and shape-based), while adhering to six restrictions on datasets and methods to make the comparison as unbiased as possible. A phased evaluation approach was then designed for summarizing dataset-level assessment metrics and discussing the results. The benchmark study presented can be a useful reference for the research community on its own; and the dataset-level assessment metrics reported may be used for designing evaluation frameworks to answer different research questions.

1. Introduction

A time series is a sequence of variable values ordered by time. These data are analyzed using a variety of statistical techniques, such as classification, clustering, and anomaly detection. This paper focuses on clustering. Clustering is a well-known unsupervised machine learning method for dividing data points (i.e., observations) into groups (called "clusters") such that observations within the same cluster tend to be more similar (according to a pre-specified criteria) than those in different clusters (Wu & Kumar, 2009). Time series data and its clustering applications abound in many disciplines. Examples include financial portfolio building (Iorio et al., 2018) and enhanced index tracking (Gupta & Chatterjee, 2018) using financial data, personalized drug design (Pirim et al., 2012) and cancer sub-type identification (Souto et al., 2008) using gene expression data, watershed management and conservation efforts (Bende-Michl et al., 2013; Dupas et al., 2015; Javed et al., 2019; Mather & Johnson, 2015; Minaudo et al., 2017) using environmental sensor-generated sample data, and anomaly detection (Flanagan et al., 2017) using network traffic data.

With the increasing prevalence of time series data, time series clustering has been gaining much attention over the past decade in order to identify previously unknown trends (Aghabozorgi et al., 2015; Begum et al., 2015; Du et al., 2019; Paparrizos & Gravano, 2016, 2017). The evaluation of clustering algorithms, however, is inherently challenging because these statistical algorithms are, by design, exploratory in nature. For this reason, the algorithm evaluation must rely on empirical study, essentially assessing how well the algorithm "rediscovers" already known classifications (Begum et al., 2015; Paparrizos & Gravano, 2016, 2017) of a given time series data.

The University of California (UCR) time series archive (Dau, Keog et al., 2018) is arguably the most popular and largest labeled time series data archive, with thousands of citations and downloads. At the time of this writing, the archive had a total of 128 datasets comprising a variety of synthetic, real, raw and pre-processed data. The archive was originally born out of frustration, with *classification* research papers reporting error rates on a single time series dataset and implying that the results would generalize to other datasets. In order to standardize the evaluation of algorithms, each dataset in the UCR archive has been split into training and test data. Additionally, each dataset is accompanied by three baseline straw man classification accuracy scores obtained using the K-nearest neighbor algorithm and different input parameter settings (window size) for dynamic time warping (DTW) (Sakoe & Chiba, 1978).

Despite extensive use of the archive in creating, validating and evaluating some of the most recently popular time series clustering algorithms (Begum et al., 2015; Paparrizos & Gravano, 2016, 2017), at the time of this writing, the archive provides no equivalent *assessment metrics* for assisting with evaluation or validation of the clustering algorithms. The latter is the single largest limitation of the archive when used for assessing clustering algorithms. Different researchers must repeat the process of implementing and benchmarking clustering algorithms over the same data sets. At a minimum, this may cost months or longer of run time (Paparrizos & Gravano, 2017); and when

* Correspondence to: 82 University Place, Innovation Hall, Burlington, VT 05405, USA.
E-mail addresses: ali.javed@uvm.edu (A. Javed), bslee@uvm.edu (B.S. Lee), drizzo@uvm.edu (D.M. Rizzo).

https://doi.org/10.1016/j.mlwa.2020.100001
Received 5 May 2020; Received in revised form 28 June 2020; Accepted 6 July 2020
Available online 23 July 2020

benchmark tests are repeated, the subjective nature of test details (e.g., pre-processing) may introduce bias that affects the objectivity and re-producibility of the test results.

The work presented in this paper aims to address the limitation associated with testing time series clustering algorithms by providing a clustering benchmark. The intent of this benchmark is similar to the classification benchmark of Dau, Keog et al. (2018), that is to provide comparison with several established methods in order to reduce both the repetition of experiments and time to publication. We would add to this another goal, that is to study the impact of changing design choices that occur within a given clustering method (i.e., a combination of clustering algorithm and distance measure). Additionally, the discussion highlights the value of considering a pool of clustering methods for use in cluster analysis and provides guidance on how to select individual algorithms in such a pool. To this end, we select eight clustering methods in this benchmark study that span three types of clustering algorithms and three distance measures, and assess each while adhering to the six restrictions laid out below.

1. *No pre-processing.* All datasets in the archive were used without any additional pre-processing (e.g., normalization in magnitude, filtering, smoothing). The reason is that, while pre-processing is common and is shown to improve results (Rakthanmanon et al., 2012), any improvement resulting from the pre-processing should not be attributed to the clustering method itself (Dau, Keog et al., 2018; Keogh & Kasetty, 2003) and, even if it were, the same pre-processing may have different performance impacts on different clustering methods.
2. *Only uniform length time series.* Only datasets in which all time series have equal length are used. The reason is that some of the clustering methods used in this benchmark were designed to work only with time series of equal length. (Only 11 out of 128 datasets in the archive have varying time series length.)
3. *Known number of clusters.* The clustering methods used in this work require that the number of clusters, k, be provided as input. The value of k is known from the class labels annotated in the datasets. There are several techniques for estimating k (e.g., Bezdek & Pal, 1998; Bholowalia & Kumar, 2014; Patil & Baidari, 2019; Subbalakshmi et al., 2015), but evaluating those techniques is not part of this benchmark.
4. *Minimum two classes.* Only datasets with $k = 2$ or more classes (other than a class designated as "noise") are used, as clustering time series data that all belong to the same class (i.e., $k = 1$) is not meaningful. (Five datasets have less than two classes.)
5. *Established methods.* All clustering methods used in this work are well-established or have survived the test of time. They are treated with equal merit with no effort to identify one as "superior" or "inferior" to another.
6. *Dataset-level assessment metrics.* The assessment metrics are reported for each clustering method on each of the 112 remaining datasets. Using assessment metrics at the dataset level enables evaluation frameworks to be designed with the research questions in mind, eliminating repetitive experimentation.

The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 describes the benchmark methods. Section 4 presents the benchmark test results. Section 5 highlights the limitations and related opportunities of the benchmark. Section 6 concludes the paper.

2. Related work

Benchmarking, in general, has been recognized as an important step in advancing the knowledge of both supervised and unsupervised learning (Ding et al., 2010; Fränti & Sieranoja, 2018; Keogh & Kasetty, 2003; Mechelen et al., 2018). See Keogh and Kasetty (2003) for a nice summary on the need to benchmark time series algorithms. They highlight many studies that use straw man algorithms to compare time series classification algorithms, and note that many of these algorithms provide little value because the levels of improvement are completely dwarfed by the variance observed when tested on real datasets or when minor unstated implementation details change. After a thorough survey of more than 350 time series data mining papers, they concluded that a median of only 1.0 (or an average of 0.91) rival methods were compared against a "novel" method (e.g., clustering algorithm, distance measure, pre-processing); and on average, each method was tested on only 1.85 datasets. While their summary is based on time series *classification*, the same concerns apply to time series *clustering*.

Works that compare time series clustering methods suggest that these comparisons have either been done qualitatively, using a theoretical approach (e.g., Ali et al., 2019; Liao, 2005; Roddick & Spiliopoulou, 2002), or quantitatively using an empirical approach (e.g., Begum et al., 2015; Paparrizos & Gravano, 2016, 2017). Only the empirical approaches provide evidence of performance measured on external datasets. The UCR archive has been used for that purpose in most of the recent time series clustering comparisons (e.g., Begum et al., 2015; Paparrizos & Gravano, 2016, 2017). However, none of them reports assessment metrics at the dataset level accounting for all datasets in the archive because the goal was to evaluate a novel method in the context of unique research questions/objectives. While it may serve individual research goals, the summarized results are often difficult and time-consuming to re-produce because of missing details (e.g., parameter settings, pre-processing details) and non-deterministic nature of the algorithm (e.g., K-means).

The absence of assessment metrics at the dataset level means that researchers must repeat experiments in order to view the tradeoffs among methods, thereby wasting precious resources and often delaying publications. The benchmark provided in this paper is intended to relax some of the burdens on researchers to foster more objective benchmark studies.

Table 1
Eight benchmark clustering methods. [1] (Paparrizos & Gravano, 2016), [2] (Sakoe & Chiba, 1978), [3] (Du et al., 2019), and [4] (Begum et al., 2015).

Clustering Method		Category
Clustering algorithm	**Distance measure**	
K-means	Euclidean	Partitional
K-medoids	Euclidean	
Fuzzy C-means	Euclidean	
K-means	Shape-based [1]	
K-means	DTW [2]	
Density Peaks [3]	Euclidean	Density-based
Density Peaks	DTW (TADPole [4])	
Agglomerative	Euclidean	Hierarchical

3. Benchmark methods

The benchmark methods comprise clustering methods (Section 3.1) and evaluation methods (Section 3.2).

3.1. Clustering methods

There are two major design criteria in clustering methods: the clustering algorithm and the distance measure. Eight clustering methods are used in this benchmark (see Table 1). They represent three categories of clustering algorithms – partitional, density-based, and hierarchical – and three distance measures — Euclidean, dynamic time warping (DTW), and shape-based. This subsection summarizes the clustering algorithms and distant measures.

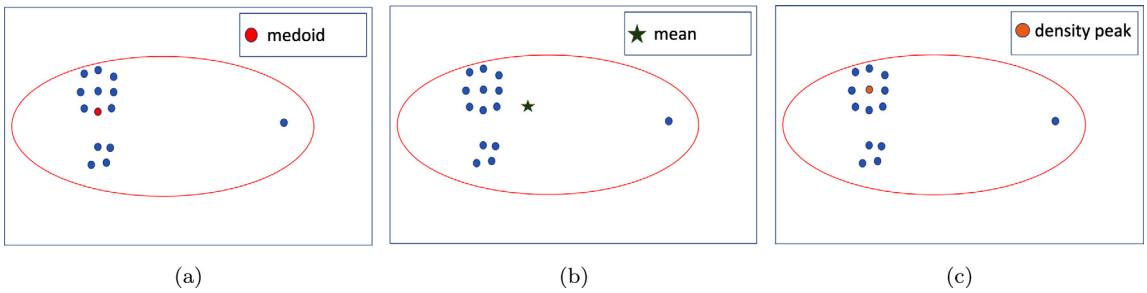


Fig. 1. Different types of centroids: (a) medoid in K-medoids, (b) centroid in K-means, and (c) density peak in Density Peaks.

3.1.1. Clustering algorithms

Choice of clustering algorithms may depend on the strategy used to maximize the intra-group similarity and minimize the inter-group similarity. The algorithms considered in this benchmark cover three popularly used categories of such strategies, each described below.

Partitional

Three partitional clustering algorithms, K-means (MacQueen, 1967), K-medoids (Kaufman & Rousseeuw, 1990), and Fuzzy C-means (Bezdek, 1981), are selected based on their popularity (Ali et al., 2019) and known accuracy for time series data clustering (Paparrizos & Gravano, 2017). Note K-means with shape-based distance is K-shape (Paparrizos & Gravano, 2017). These partitional algorithms generate spherical clusters that are similar in size (Liao, 2005); and optimize clustering by minimizing the distance between each cluster center (a.k.a. centroid) and the data points within that cluster. A centroid may or may not be an actual data point, depending on the algorithm — it is for K-medoids and not for K-means and Fuzzy C-means (see Figs. 1(a) and 1(b)).

All three of these partitional algorithms require that one input parameter be specified — the number of clusters (k). Given k, the algorithm iterates over two phases: (1) calculate centroids, and (2) assign data points to their closest centroid, until some termination condition (e.g., number of iterations or convergence) is met. For all three algorithms used in this benchmark, the initial centroids are chosen at random, making the algorithm non-deterministic; all subsequent centroids are calculated so as to minimize the distance to all other data points within the given cluster.

While K-means and K-medoids are hard clustering algorithms (i.e., producing non-overlapping partitions), Fuzzy C-means is a soft clustering algorithm (i.e., producing overlapping partitions). In this benchmark, the Fuzzy C-means clustering results are similar to that of a hard clustering algorithm, as each data point is assigned to the cluster that has the highest probability. There are several techniques for improving the clustering accuracy of these algorithms including — performing z-score normalization[1] on the input (Mohamad & Usman, 2013), or invoking the algorithm multiple times using different random seeds to select the clusters with the highest intra-cluster similarity and the lowest inter-cluster similarity. This benchmark excludes using such techniques, per restrictions 1 and 5 (see Section 1).

Density-based

Density Peaks (Du et al., 2019) was selected as the representative for density-based algorithms due to its recent popularity, particularly for time series clustering (Begum et al., 2015). Unlike other density-based algorithms (Ester et al., 1996), Density Peaks is not sensitive to the "density parameter" but needs the number of clusters, k, as one of the inputs. This makes it a good fit for this benchmark, where k is assumed to be known and no assumptions are made for other input parameters.

The Density Peaks algorithm generates cluster centroids (called "density peaks") that are surrounded by neighboring data points that have lower local density (see Fig. 1(c)) and are relatively farther from data points with a higher local density (Du et al., 2019). The algorithm has two phases. It first finds centroids (density peaks), and then assigns data points to the closest centroid. The algorithm requires two input parameters: the number of clusters (k) and the local neighborhood distance d (wherein the local density of a data point is calculated). While the value of k is assumed to be known in this benchmark, the value of d is determined as the distance wherein the average number of neighbors is 1 to 2% of the total number of observations in the dataset, following a rule of thumb proposed by the original authors (Rodriguez & Laio, 2014).

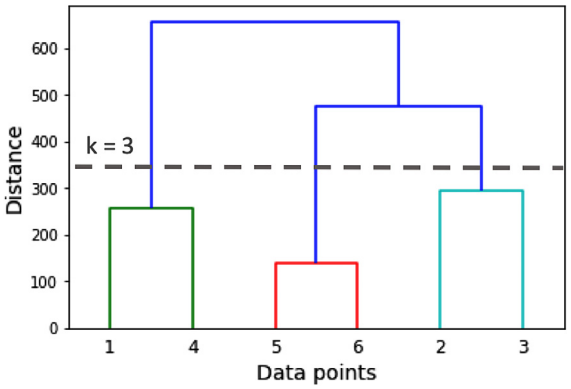


Fig. 2. Agglomerative clustering.

Hierarchical

A hierarchical clustering algorithm can be Agglomerative (bottom-up) or divisive (top-down). In the former, each data point begins as its own cluster and cluster pairs are merged as the algorithm moves up the hierarchy. In the latter, all data points are initially assigned to a single cluster and clusters are split as the algorithm moves down the hierarchy. Because of its popularity over divisive clustering (Liao, 2005), Agglomerative clustering is used in this benchmark.

The algorithm has two phases. It first initializes each data point into its own cluster and then repeatedly merges the two nearest clusters into one until there are k clusters (see Fig. 2). The value of k is an input to the algorithm. There are several options for measuring the distance between pairs of clusters. Ward's linkage, which minimizes the variance of data points in the merged clusters (Großwendt et al., 2019), is used in this benchmark due to its popularity and also its similarity to the optimization strategy of the partitional clustering methods. Other popular distance measures include single-linkage (minimum distance between a pair of data points belonging to different clusters) and complete-linkage (maximum distance between a pair of data points belonging to different clusters) (Li & de Rijke, 2017).

[1] About 80% of datasets in the UCR archive are z-score normalized.

3.1.2. Distance measures

The choice of distance measure is the other criterion that has a direct impact on the clustering performance. This section discusses the three distance measures used in this benchmark.

Euclidean distance

The most common distance measure used in a broad range of application is the Euclidean distance (Faloutsos et al., 1994). Eq. (1) shows how the Euclidean distance $d(T1, T2)$ is calculated between two time series $T1 = (T1_1, T1_2, \dots, T1_n)$ and $T2 = (T2_1, T2_2, \dots, T2_n)$.

$$d(T1, T2) = \sqrt{\sum_{i}^{n} (T1_i - T2_i)^2} \tag{1}$$

Dynamic time warping

Dynamic time warping (DTW) is a mapping of points between a pair of time series, $T1$ and $T2$ (see Fig. 3) designed to minimize the pairwise Euclidean distance. It is becoming recognized as one of the most accurate similarity measures for time series data (Johnpaul et al., 2020; Paparrizos & Gravano, 2017; Rakthanmanon et al., 2012). The optimal mapping should adhere to three rules.

- Every point from $T1$ must be aligned with one or more points from $T2$, and vice versa.
- The first and last points of $T1$ and $T2$ must align.
- No cross-alignment is allowed, that is, the warping path must increase monotonically.

DTW is often restricted to mapping points within a moving window. In general, the window size could be optimized using supervised learning with training data; this, however, is not possible with clustering as it is an unsupervised learning task. Paparrizos and Gravano (2016) found 4.5% of the time series length to be the optimal window size when clustering 48 of the time series datasets in the UCR archive; as a result, we use a fixed window size of 5% in this benchmark study.

Density Peaks with DTW as the distance measure can be computationally infeasible for larger datasets because the Density Peaks algorithm is non-scalable of $O(n^2)$ complexity (Paparrizos & Gravano, 2017). We employ a novel pruning strategy (see TADPole (Begum et al., 2015) to speed up the algorithm by pruning unnecessary DTW distance calculations.

Shape-based distance

Shape-based distance is both shift-invariant and scale-invariant (Paparrizos & Gravano, 2016), that is, not affected by the shifting or scaling of the time series data. It calculates the cross-correlation between two time series and produces a distance value between 0.0 to 2.0, with 0.0 indicating that the time series are identical and 2.0 indicating maximally different shapes. To ensure the distance measure is scale-invariant, each original time series, T, is z-normalized to T' as follows (Paparrizos & Gravano, 2016):

$$T' = \frac{T - \mu}{\sigma} \tag{2}$$

so T' has mean $\mu' = 0$ and standard deviation $\sigma' = 1$.

3.2. Evaluation methods

The purpose of this benchmark study is to assess the performance of the eight clustering algorithms on the 112 datasets, as well as the impact of changing design choices in either clustering algorithms or distance measures. To this end, the evaluation framework and select assessment metrics are discussed in this section.

3.2.1. Assessment metrics

Metrics for assessing clustering output may be external or internal. External measures are used when the class labels are available for individual data points. Examples include the Rand Index (RI) (Hubert & Arabie, 1985), Adjusted Rand Index (ARI) (Santos & Embrechts, 2009), Adjusted Mutual Information (AMI) (Romano et al., 2016), Fowlkes Mallows index (FMS) (Fowlkes & Mallows, 1983), Homogeneity (Rosenberg & Hirschberg, 2007), and Completeness (Rosenberg & Hirschberg, 2007). Internal measures quantify the goodness of clusters based on a optimization objective for the clustering output, without the need for class labels; examples include Silhouette score (Rousseeuw, 1987), Davies–Bouldin index (Davies & Bouldin, 1979), Calinski–Harabasz index (Caliński & J.A., 1974), the I-index (Maulik & Bandyopadhyay, 2002) and sum of square errors (SSE).

We used all the external measures listed above in this benchmark because having the class labels provided in the UCR archive makes the evaluation independent of the algorithm's optimization function. Despite the popularity of the Rand Index (Fig. 4(f)) for prior UCR archive studies (e.g., Begum et al., 2015; Paparrizos & Gravano, 2016, 2017), we find the adjusted measures more suitable for clustering because they are independent of the number of clusters. As demonstrated in Fig. 4, the accuracy scores resulting from random cluster assignment are consistently low as the number of clusters varies for the two adjusted measures (Figs. 4(a) and 4(b)), while this is not the case for the other measures. In this work, the Adjusted Rand Index was selected as the default measure unless stated otherwise.

For the partitional algorithms in this benchmark, all of which are non-deterministic, the scores reported for each external measure are the average over ten runs using randomly selected initial centroids.

Adjusted Rand Index

The Adjusted Rand Index is the adjusted-for-chance version of the more commonly used Rand Index. Given two sets of clusters, X and Y, and a contingency table where each cell n_{ij} is the number of elements in both the ith cluster of X and the jth cluster of Y, the Adjusted Rand Index is calculated as shown in Eq. (3).

$$\text{Adjusted Rand Index} = \frac{\sum_i^j \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{3}$$

where a_i is the sum of the ith row and b_j is the sum of the jth column in the contingency table.

Spread between clustering outputs

The measure of spread is used to quantify how much the accuracy of the two clustering methods differ from each other over multiple datasets (see Eq. (4)).

$$\text{Spread} = \frac{\sum_{i=1}^{n} (A1_i - A2_i)^2}{n} \tag{4}$$

where $A1_i$ and $A2_i$ are the accuracy scores of the two methods for dataset i; and n is the total number of datasets.

3.2.2. Evaluation framework

Researchers will often design an evaluation framework for assessing accuracy because what constitutes "good" with respect to the assessment metrics may vary depending on the research question. One of the simplest approaches is to rank the performance of each clustering method and tally the number of winning performances across all available (in this work 112) datasets. This approach, however, is not without bias, as it depends on the distribution of both the datasets and clustering methods. For instance, in this work there are five partitional methods and one density-based method. If one half the datasets are amenable to partitional and the other half to density-based, this evaluation metric will bias the density-based method because the tally for the partitional methods would be partitioned across the five datasets.

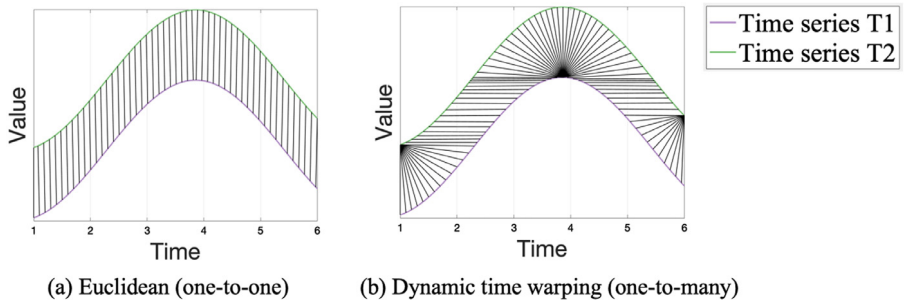


Fig. 3. Alignment between two times series for calculating distance.

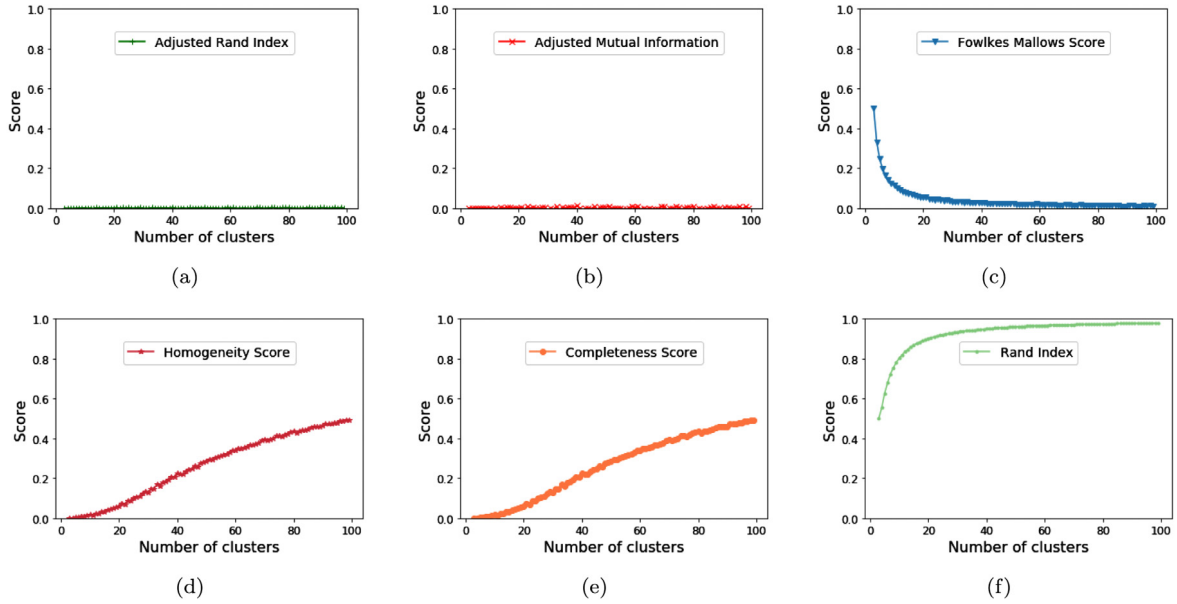


Fig. 4. Accuracy scores resulting from randomly assigning 1000 data points to a varying number of clusters.

On the other extreme, if pairwise comparison were performed on all clustering methods, it would result in 28 ($= \binom{8}{2}$) pairwise comparisons for each of the 112 datasets (i.e., 3136 comparisons). More importantly, a pairwise comparison assumes that every algorithm is designed to achieve the same result.

Based on the above challenges, we designed a phased evaluation approach in this benchmark study. This approach first compares the eight clustering methods, and then controls for either the distance measure or clustering algorithm while evaluating the impact of changing the other.

Phase 1. All eight methods are compared using all datasets, and the resulting accuracy is averaged over all datasets for each method.

Phase 2. Partitional algorithms with Euclidean distance are compared to select the one that achieves the highest accuracy on the largest number of datasets.

Phase 3. Different distance measures are compared using the partitional algorithm selected in Phase 2.

Phase 4. Clustering algorithms belonging to different categories are compared using Euclidean distance. Among them, the partitional algorithm is the one selected in Phase 2 (i.e., K-means with Euclidean distance).

Phase 5. Density Peaks algorithm using Euclidean distance is compared with Density Peaks algorithm using DTW.

Phase 6. Density Peaks algorithm using DTW is compared with the partitional algorithm selected in Phase 2 but using DTW.

In Phase 1, we report the average scores and standard deviations across all datasets for all six external assessment metrics used in this work. In each subsequent phase, we report the number of datasets (called "winning count") for which an algorithm or a distance measure achieved the highest ARI, and refine the comparison with the measure of spread (see Section 3.2.1) and the associated scatter plots. Here, datasets that result in an ARI score lower than 0.05 are excluded from winning counts since scores that approach 0.00 represent random assignment.

4. Benchmark test results

This section provides the results of dataset-level assessment (Section 4.1) and the phased evaluation (Section 4.2), and discusses the results (Section 4.3).

4.1. Dataset-level assessment

The Appendix shows the Adjusted Rand Index (ARI) scores for all eight clustering methods on the 112 short-listed datasets (see Section 1) in the UCR archive (Table A.1), and the spread of ARI scores (Table A.2) between each pair of clustering methods. Additionally, in line with the restriction 6 (dataset-level assessment; see Section 1), the scores of each clustering method on each dataset tested for all the six external measures (see Section 3.2.1) are available at GitHub (Javed, 2019) along with the source codes.

Table 2
Average and standard deviation of adjusted measures for each clustering method in Phase 1.

Clustering Method		Category	ARI		AMI	
Algorithm	**Distance measure**		Avg	Std	Avg	Std
Agglomerative	Euclidean	Hierarchical	0.26	0.26	0.31	0.27
K-means	DTW	Partitional	0.24	0.24	0.29	0.25
K-means	Euclidean		0.24	0.24	0.29	0.24
Fuzzy C-means	Euclidean		0.22	0.25	0.24	0.25
K-medoids	Euclidean		0.22	0.23	0.26	0.25
K-means	Shape-based		0.21	0.22	0.25	0.23
Density Peaks	Euclidean	Density-based	0.19	0.24	0.25	0.26
Density Peaks	DTW		0.16	0.25	0.24	0.27

Table 3
Average and standard deviation of non-adjusted measures for each clustering method in Phase 1.

Clustering Method		RI		Homogeneity		Completeness		FMS	
Algorithm	**Distance measure**	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Agglomerative	Euclidean	0.72	0.17	0.34	0.28	0.36	0.29	0.51	0.20
K-means	DTW	0.71	0.16	0.31	0.27	0.34	0.28	0.51	0.19
K-means	Euclidean	0.72	0.16	0.32	0.25	0.33	0.27	0.49	0.19
Fuzzy C-means	Euclidean	0.69	0.15	0.27	0.26	0.31	0.27	0.48	0.21
K-medoids	Euclidean	0.71	0.15	0.30	0.25	0.31	0.25	0.47	0.19
K-means	Shape-based	0.66	0.17	0.27	0.23	0.38	0.29	0.50	0.18
Density Peaks	Euclidean	0.65	0.18	0.27	0.26	0.34	0.29	0.50	0.20
Density Peaks	DTW	0.62	0.18	0.25	0.26	0.36	0.31	0.51	0.20

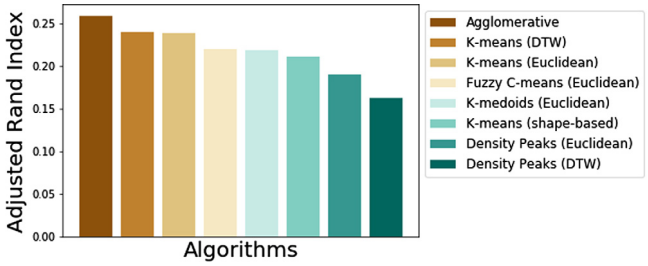


Fig. 5. Average ARI for each clustering method in Phase 1.

Table 4
Clustering algorithms with Euclidean distance in Phase 2.

Algorithm	Winning count
Triple-wise	
K-means	54
Fuzzy C-means	31
K-medoids	18
Pairwise	
K-means	64
K-medoids	17
K-means	54
Fuzzy C-means	27
Fuzzy C-means	41
K-medoids	39

Table 5
Different distance measures for K-means (from Phase 2) in Phase 3.

Distance measure	Winning count
Triple-wise	
DTW	32
Shape-based	31
Euclidean	28
Pairwise	
DTW	45
Euclidean	38
DTW	52
Shape-based	38
Shape-based	45
Euclidean	44

4.2. Phased evaluation

Phase 1 - Ranked comparison of all methods. Fig. 5 shows the average ARI's for each of the eight clustering methods in decreasing order. In addition, Tables 2 and 3 provide details, including the average and standard deviation of the clustering scores resulting from the six external assessment metrics (see Section 3.2.1). Table 2 shows the results for the two adjusted metrics ARI and AMI; they are in agreement about the highest and lowest scorers in terms of the average score across all datasets. The highest average was for the Agglomerative clustering using Ward linkage and Euclidean as distance measure; and the lowest average was for Density Peaks using DTW as distance measure.

Table 3 shows the results for the other (non-adjusted) metrics, RI, Homogeneity, Completeness, and FMS. They result in ordering of scores different from the ordering from the (adjusted) ARI and AMI. Since those measures are not independent of the value of k, averaging their scores across datasets with different k values is not so meaningful in this benchmark. For instance, for certain datasets such as GunPointAgeSpan, GunPoindMaleVersusFemale and GunPointOldVersusYoung (see Appendix), K-means with shape-based distance converged to a single cluster during the iterative process, thus maximizing the Completeness score to 1.0 (for k=1), and keeping the FMS score higher than it would be for $k > 1$; in contrast, this convergence to $k = 1$ penalizes K-means with shape-based distance when Homogeneity is used for scoring the result. Like this, these non-adjusted measures are driven to be biased toward extreme values of k (i.e., 1 or the number of data points) and consequently should not be used for averaging the scores from datasets with different k values.

The standard deviations shown in Tables 2 and 3 are rather significant relative to the average values for all assessment metrics used. This indicates the wide variation of the scores across different datasets.

Phase 2 - Comparison of partitional algorithms using Euclidean distance. Of the partitional clustering methods that use a Euclidean distance measure, K-means had a winning count of 54 datasets, while Fuzzy C-means and K-medoids performed best on 31 and 18 datasets, respectively, (see Table 4). While K-means had a higher ARI score in almost twice as many datasets, differences in score values were minor, with a

Table 6
Different algorithms with Euclidean distance measure in Phase 4.

Algorithm	Winning count
Triple-wise	
Agglomerative	45
K-means	21
Density Peaks	19
Pairwise	
Agglomerative Density Peaks	57 26
Agglomerative K-means	52 30
K-means Density Peaks	60 23

Table 7
Euclidean vs. DTW for Density Peaks algorithm in Phase 5.

Distance measure	Winning count
Euclidean	45
DTW	31

Table 8
DTW in Density Peaks and K-means (selected in Phase 2) in Phase 6.

Algorithm	Winning count
K-means	60
Density Peaks	24

spread of only 0.005 against K-medoids (see Fig. 6(a)) and only slightly larger (0.010) against Fuzzy C-means (see Fig. 6(b)). This result is not surprising, given the similarity of methodology (all partitional using Euclidean distance) across the three algorithms.

Phase 3 - Comparison of distance measures using selected partitional algorithm. When we examine the winning counts for K-means (i.e., method that performed best in Phase 2) using the three distance measures, the tallies are 32, 31 and 28 for DTW, shape-based, and Euclidean, respectively (see Table 5). A pairwise comparison between the distance measures also shows the wining counts to be 45 vs. 38 between DTW and Euclidean, 52 vs. 38 between DTW and shape-based, and 45 vs. 44 between shape-based and Euclidean. The scatter plots in Fig. 7 show the spreads between each of the paired distance measures. The shape-based distance has a relatively larger spread with each of the other two measures. As a side note, when the optimal DTW window size is assumed to be known, then it is trivial to understand that DTW will always achieve a score that is higher or equal to that of Euclidean distance, since the two measures are equivalent when the window size is 0.

Phase 4 - Comparison of clustering algorithms using Euclidean distance. When we hold the distance measure (in this case, Euclidean distance) constant and examine the winning counts across the clustering algorithms that use this distance measure, the tallies are 45, 21, and 19 in the order of Agglomerative, K-means, and Density Peaks. A pairwise comparison is also shown in Table 6, where the winning counts are 57 vs. 26 between Agglomerative and Density Peaks, 52 vs. 30 between Agglomerative and K-means, and 60 vs. 23 between K-means and Density Peaks. Despite the difference in winning counts, the spreads of ARI values between Agglomerative and K-means (see Fig. 8(a)) is fairly small compared with the spread of either method with Density Peaks (see Fig. 8(b) and Fig. 8(c)).

Phase 5 - Comparison of Euclidean distance and DTW in Density Peaks algorithm. The Density Peaks algorithm achieved a higher winning count (i.e., across 45 datasets; see Table 7) when Euclidean distance was used as the distance measure compared to a count of 31 with DTW. Fig. 9 shows the spread of ARI scores between Euclidean distance and DTW to be 0.021.

Phase 6 - Comparison of Density Peaks and selected partitional algorithm using DTW. Lastly, when the DTW distance measure is held constant, we may compare across the clustering algorithms that use this distance measure — Density Peaks and K-means. K-means achieved a higher winning count (i.e., winner across 60 datasets; see Table 8) compared to a winning count of 24 for Density Peaks. But while the winning count appears positively skewed in favor of K-means, there are still a considerable number of datasets for which Density Peaks achieved higher ARI, and the spread of ARI scores (see Fig. 10) was the largest (0.052) observed in the six phases.

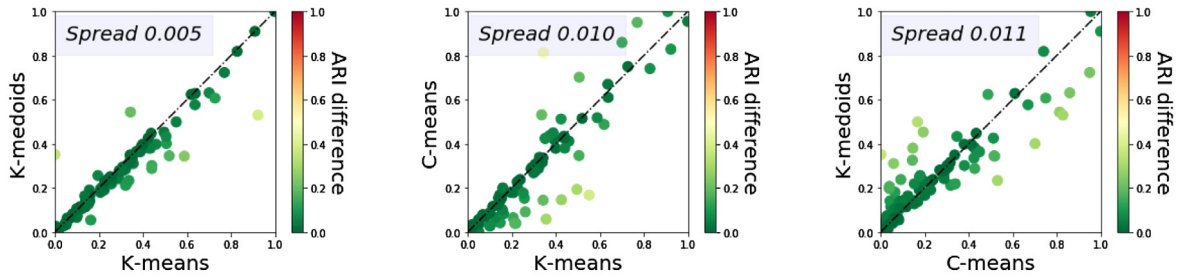
4.3. Discussion

This section analyzes the results of each evaluation phase and provides concluding remarks summarizing the analysis.

Phase 1 - Ranked comparison of all methods. The high standard deviations associated with the average scores of Tables 2 and 3 suggest that accuracy is dependent on which clustering method is used on which dataset; and that it may be fair to conclude that we have no clear winner in this benchmark. The high variability in scores also suggests that using a simple winning count of dataset-level assessment as the only means of evaluation, may be very misleading. While reporting counts of win–lose–tie for clustering method accuracy has become common practice in the literature, the UCR archive authors describe it as not that useful (Dau, Keog et al., 2018). In light of these issues as well as noting that adjusted measures are more suitable in this benchmark, we used both winning counts and the ARI scores in this benchmark and reinforced the measures with ARI score scatter plots and the associated spreads.

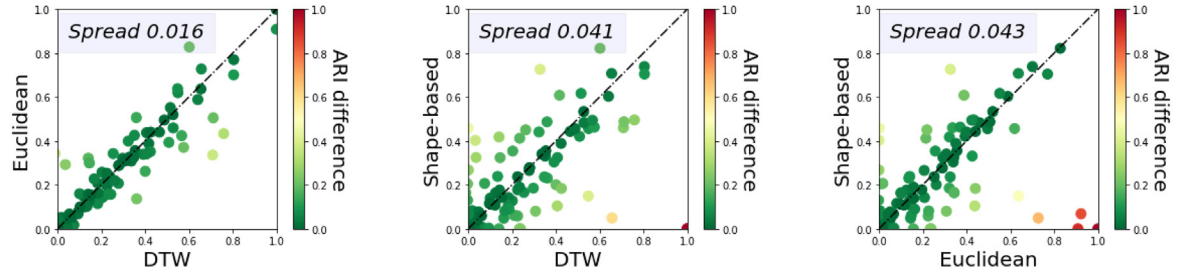
Phase 2 - Comparison of partitional algorithms using Euclidean distance. When comparing the three partitional algorithms that use the Euclidean distance measure, a researcher may well select K-means based on the winning count (see Table 4), especially without adequate prior knowledge of how the algorithm performs on the individual datasets. However, the selection may likely change when the user has knowledge of the dataset and/or application at hand. For instance, K-medoids is more resilient to outliers, because the medoids are not as sensitive to the presence of outliers as say, the centroids in K-means. In another example, Fuzzy C-means may be preferred over K-means given a dataset where the membership of data points are "soft", as in the case when categorical classes have numerical attribute values that overlap. As an aside, Fuzzy C-means shows a larger spread of ARI scores against K-means (Fig. 6(b)) and K-medoids (Fig. 6(c)), indicating that changing from K-means to the fuzzy mechanism of C-means has more impact on the final clustering than changing from means to medoids.

Phase 3 - Comparison of distance measures using selected partitional algorithm. The results in Table 5 appear to suggest that the winning count does not favor the shape-based distance measure in the same manner that it did in a prior study (Paparrizos & Gravano, 2017) that used 85 datasets in the UCR archive compared to the 112 datasets (and different evaluation criteria) used in this benchmark study. The larger spreads observed when one distance measure is shape-based (Figs. 7(b) and 7(c)) suggest the method is useful as the best distance measure for a nontrivial number of datasets, and therefore, should be considered in a pool of potential clustering methods. We believe the larger spread may be a result of the shape-based distance measure's lack of sensitivity to the magnitudes and shifts in time series data compared with the Euclidean measure, or for that matter, DTW (for which the underlying distance measure is also Euclidean), which therefore results in a different partitioning.



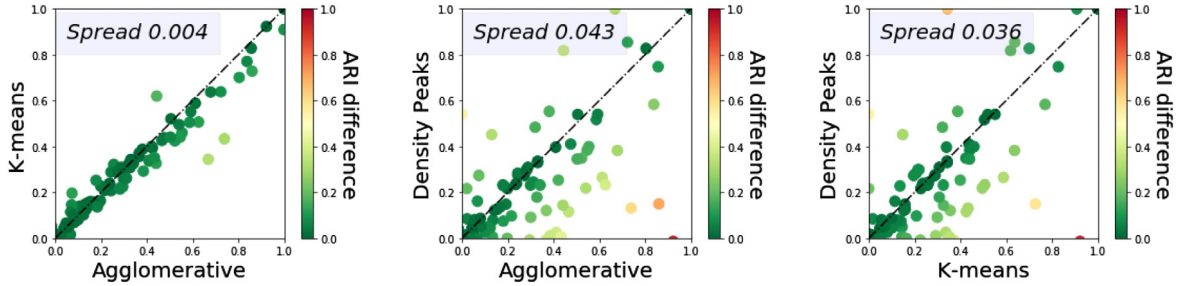
(a) K-medoids vs. K-means. (b) Fuzzy C-means vs. K-means (c) K-medoids vs. Fuzzy C-means.

Fig. 6. Spread of ARI scores between each pair of the three clustering algorithms with Euclidean distance in Phase 2.



(a) Euclidean vs. DTW (b) Shape-based vs. DTW (c) Shape-based vs. Euclidean

Fig. 7. Spread of ARI scores between each pair of distance measures in Phase 3.



(a) K-means vs. Agglomerative. (b) Density Peaks vs. Agglomerative. (c) Density Peaks vs. K-means.

Fig. 8. Different algorithms with Euclidean distance measure in Phase 4.

Phase 4 - Comparison of clustering algorithms using Euclidean distance. The very small spread in Fig. 8(a) shows similar performance for the K-means and Agglomerative algorithms on most datasets in the archive. With Agglomerative clustering, this can be attributed to the use of Ward's linkage, which merges the two clusters that when combined provide the minimum increase in variance. This optimization using Ward's linkage has some similarity to optimizing the centroids in K-means (i.e., minimizing the total variance within cluster). Using a different linkage criteria such as "complete" linkage does not bias clusters to be as spherical as Ward linkage (and for that matter K-means). Such a change will result in different clusters when compared to K-means. Specifically, with complete linkage, Agglomerative clustering has a measure of spread of 0.026 when compared to K-means, and an average ARI of 0.17 ± 0.24.

Phase 5 - Comparison of Euclidean distance and DTW in Density Peaks algorithm. The spread (0.021) between DTW and Euclidean (see Fig. 9) in Density Peaks algorithm is relatively consistent with spread (0.016) between DTW and Euclidean in K-means algorithm (see Fig. 7(a)). These medium to high level of spread values indicate the difference of clusters formed when using DTW as opposed to Euclidean distance. Density Peaks is an $O(n^2)$ complexity algorithm (where n is the number of data points) that when used with DTW may become computationally infeasible for large datasets. The TADPole method (Begum et al., 2015), with its novel pruning strategy, makes Density Peaks with DTW feasible enough for use on large datasets in the archive. However, even with this accelerated TADPole, the largest 20 datasets of the archive took 32 days to cluster on a dual 20-Core Intel Xeon E5-2698 v4 2.2 GHz machine with 512 GB 2133 MHz DDR4 RDIMM.

Phase 6 - Comparison of Density Peaks and selected partitional algorithm using DTW. When using DTW as a distance metric, K-means and Density Peaks produce different clusters as indicated by the relatively higher spreads of ARI 0.052 (see Fig. 10), which is consistent with the somewhat high spread 0.036 observed between the two methods (see Phase 4 with Euclidean distance, Fig. 8(c)). This result is counter-intuitive given that both K-means and Density Peaks form spherical clusters by assigning data points to the closest centroid, and leads one to speculate that the cause may be the fundamentally different locations of the centroids in the K-means and Density Peaks algorithms (see Fig. 1).

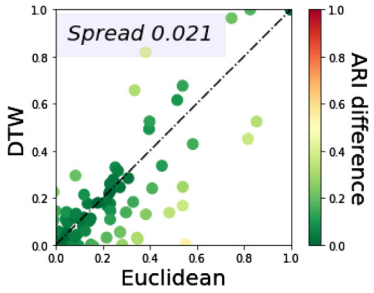


Fig. 9. Euclidean vs. DTW for Density Peaks algorithm in Phase 5.

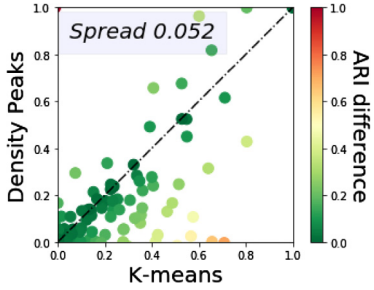


Fig. 10. DTW in Density Peaks and K-means (selected in Phase 2) in Phase 6.

Concluding remarks. Overall, this benchmark study shows that among all methods tested, the variation in performance, as measured by the average and standard deviation of ARI (see Table 2 and Fig. 5), is higher than the variation observed across winning counts (Table 4 to Table 8). Notably, there is no one method that performs better than the others for all datasets in this benchmark, and that method performance is much more sensitive with respect to the datasets, for a given evaluation objective (i.e., assessment metric). Similar findings for time series representation methods and distance measures were made in an earlier benchmark study using UCR archive (Ding et al., 2010). This is not to say that the recently invented algorithms or methods are of no use. K-means is the first and one of the most popular clustering methods invented in the 1950s (Kaufman & Rousseeuw, 2008), while Density Peaks algorithm and shape-based distance were invented more recently. While the later methods may not necessarily be superior to the earlier methods, the advances in time series clustering are noted in the collective improvements in their ability to correctly identify clusters. As new clustering methods are invented over the years, the clustering result, as assessed by the average of the maximum ARI scores achieved by different methods for each dataset in the benchmark, has been steadily increasing (see Fig. 11). In light of these two findings, and noting that exploratory cluster analysis typically involves trying multiple clustering methods rather than a single method to identify correct clusters, cluster analysis should be conducted by selecting a pool of methods that produce different clusters, rather than those that produce similar clusters. In other words, select methods that show greater spread (i.e., combination of average accuracy scores and their spread) rather than those with higher winning counts. Methods with higher spreads of ARI are likely to produce different clusters for the same dataset — all of which may be valid depending on the target research goal. For instance, using three algorithms with higher spread values (e.g., K-means (shape-based), Agglomerative (Euclidean) and Density Peaks (DTW) of Figs. 7(c), 8(a) and 9) on the same dataset are more likely to provide three dissimilar clustering outputs, compared to those generated using K-means (Euclidean), K-medoids (Euclidean), and Fuzzy C-means (Euclidean) (lower spread values in Fig. 6).

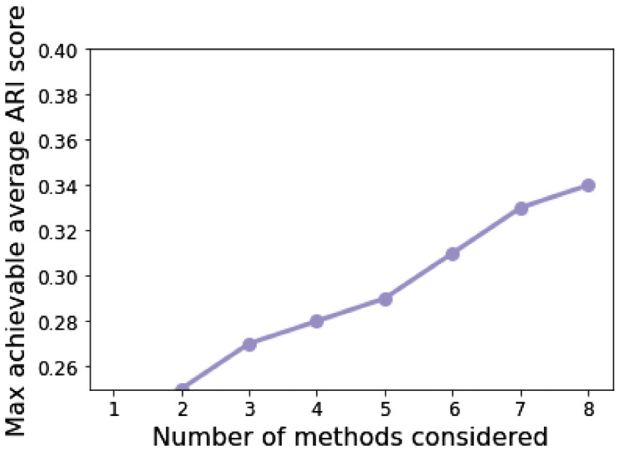


Fig. 11. Maximum achievable average ARI score for progressively increasing number of methods (over time).

5. Limitations and opportunities

There are a few managerial limitations in our benchmark that offer opportunities. First, the UCR archive is currently the best available to build a benchmark for designing and evaluating clustering algorithms. As acknowledged by the curators (Dau, Bagnall et al., 2018), however, the datasets in the archive represent the interests and hobbies of the curators, and as a result may invite a question on any benchmark built on top of the datasets. While we believe that our benchmark, built on a comprehensive set of datasets from the UCR archive, is viable for general purpose clustering methods, for specific applications it may be prudent to use in the benchmark those select datasets that are closely related to the individual applications, thus opening an opportunity for domain-specific benchmarks.

Secondly, while our benchmark helps reduce the number of clustering methods to be considered for a given dataset, deeper insights into the "mapping" between methods and datasets can help match a method to a dataset; this will be highly desirable from an application perspective. Such insights have not been adequately published, consequently leaving the application community to consider the latest method as the "state of the art". Unfortunately, the latest is not always the best choice, as this benchmark study suggests. This opens an opportunity to conduct a more in-depth study and publish the gained insights, namely dataset-method mapping for time series clustering, to meet the need.

Finally, we used only external measures to evaluate clusters in this benchmark study and it served our purpose because of the availability of class labels in the datasets. In general, however, evaluation using internal measures as an addition or alternative would open an opportunity to make the benchmark more comprehensive, especially when no class labels are available as the ground truth.

6. Conclusion

This paper reports benchmark test from applying eight popular time series clustering methods on 112 datasets in the UCR archive. One essential goal of the benchmark is to make the results available and reusable to other researchers. In this work, we laid out six restrictions to help reduce bias. Eight popular clustering methods were selected to cover three categories of clustering algorithms (i.e., partitional, density-based, and hierarchical) and three distance measures (i.e., Euclidean, Dynamic time warping, and shape-based). The dataset-level assessment metrics are reported using six external evaluation measures. Adjusted Rand Index was selected as the default measure for discussion in this paper. A phased evaluation framework was designed such that in each phase only one of the two building blocks of a clustering method –

Table A.1
ARI scores of the eight clustering methods on the 112 datasets in the UCR archive.

Dataset name	K-mean-Euc	K-med-Euc	K-mean-shape	K-mean-DTW	C-mean-Euc	D-Peaks-Euc	D-Peaks-DTW	Agglo-Euc
ACSF1	0.16	0.17	0.14	0.10	0.20	0.13	0.06	0.15
Adiac	0.25	0.25	0.24	0.23	0.18	0.23	0.11	0.18
ArrowHead	0.20	0.26	0.18	0.23	0.18	0.27	0.25	0.07
Beef	0.15	0.14	0.11	0.12	0.17	0.05	0.09	0.07
BeetleFly	0.05	0.04	0.04	0.01	0.00	0.04	0.11	−0.02
BirdChicken	0.04	0.03	0.07	0.00	0.04	0.00	0.05	0.04
BME	0.14	0.16	0.23	0.36	0.12	0.23	0.22	0.18
Car	0.14	0.14	0.13	0.20	0.16	0.05	0.03	0.11
CBF	0.33	0.22	0.73	0.33	0.34	0.14	0.10	0.44
Chinatown	0.16	0.19	−0.05	0.24	0.18	−0.07	−0.08	0.16
ChlorineConcentration	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CinCECGTorso	0.15	0.14	0.06	0.21	0.04	0.45	0.34	0.13
Coffee	0.34	0.54	0.16	−0.01	0.81	1.00	1.00	0.67
Computers	0.00	0.00	0.07	0.00	0.00	0.00	0.01	0.00
CricketX	0.10	0.07	0.16	0.13	0.03	0.04	0.14	0.11
CricketY	0.13	0.11	0.18	0.14	0.07	0.08	0.11	0.14
CricketZ	0.10	0.07	0.16	0.13	0.03	0.05	0.14	0.12
Crop	0.31	0.28	0.08	0.31	0.28	0.18	0.18	0.33
DiatomSizeReduction	0.83	0.82	0.82	0.60	0.74	0.75	0.96	0.86
DistalPhalanxOutlineAgeGroup	0.39	0.39	0.42	0.51	0.42	−0.04	−0.02	0.42
DistalPhalanxOutlineCorrect	0.00	0.00	0.00	0.00	0.00	0.00	−0.02	0.00
DistalPhalanxTW	0.43	0.38	0.50	0.76	0.43	0.13	−0.05	0.74
DodgerLoopDay	0.23	0.23	0.08	0.17	0.20	0.22	0.18	0.20
DodgerLoopGame	0.01	0.00	0.20	0.00	0.00	0.00	0.01	0.01
DodgerLoopWeekend	0.92	0.53	0.07	−0.04	0.83	−0.01	0.09	0.92
Earthquakes	0.00	0.00	0.03	0.00	0.00	0.00	−0.09	−0.01
ECG5000	0.51	0.43	0.49	0.71	0.35	0.52	0.62	0.59
ECGFiveDays	0.00	0.00	0.40	0.03	0.00	0.22	0.03	0.02
ElectricDevices	0.16	0.05	0.09	0.19	0.08	0.00	0.14	0.20
EOGHorizontalSignal	0.21	0.20	0.14	0.18	0.18	0.10	0.00	0.22
EOGVerticalSignal	0.10	0.11	0.11	0.10	0.09	0.09	0.13	0.08
EthanolLevel	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FaceAll	0.22	0.21	0.45	0.26	0.04	0.30	0.14	0.28
FaceFour	0.32	0.29	0.42	0.14	0.29	0.48	0.14	0.32
FacesUCR	0.21	0.20	0.41	0.24	0.04	0.30	0.14	0.28
FiftyWords	0.26	0.24	0.20	0.40	0.09	0.24	0.28	0.31
Fish	0.21	0.18	0.27	0.28	0.07	0.28	0.00	0.24
FreezerRegularTrain	0.29	0.25	0.28	0.28	0.29	0.27	0.05	0.24
FreezerSmallTrain	0.29	0.24	0.28	0.28	0.29	0.27	0.05	0.27
Fungi	0.64	0.63	0.15	0.55	0.61	0.85	0.52	0.72

(*continued on next page*)

algorithm and distance measure – is varied at a time. Benchmark results show the overall performance of the eight algorithms to be similar with high sensitivity to the datasets, indicating that no method is superior to the others for all datasets. Discussion of the results helps highlight the importance of creating a pool of clustering methods with high spread in accuracy scores for effective exploratory analysis.

For practical implications of our benchmark, researchers can adopt the recommendations we made in concluding remarks (Section 4.3) as is, if they are using the same clustering methods and datasets. Otherwise (i.e., with their own methods and/or datasets), they can leverage the phased evaluation framework presented in Section 3.2.2 to conduct their own benchmark study. Either way, this benchmark can be a useful resource for exploratory clustering analysis by an application community. For the future work, we plan to expand the benchmark by adding evaluations using internal measures (one of the opportunities discussed in Section 5).

CRediT authorship contribution statement

Ali Javed: Conceptualization, Methodology, Software, Writing - original draft, Visualization, Validation. **Byung Suk Lee:** Supervision, Formal analysis, Writing - review & editing, Project administration. **Donna M. Rizzo:** Supervision, Formal analysis, Writing - review & editing, Resources.

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.

Acknowledgments

This project was supported by the grant from the Barrett Foundation and Gund Institute for Environment through a Gund Barrett Ph.D. Fellowship. This material is based upon work partially supported by the National Science Foundation under VT EPSCoR Grant No. NSF OIA 1556770. We thank Drs. Patrick J. Clemins and Scott Hamshaw, Research Assistant Professors at the University of Vermont, for support in using Vermont EPSCoR's high performance computing resources. We also thank Dr. Eamon Keogh for his invaluable feedback, and all other curators and administrators of the UCR archive without which this work would not have been possible.

Appendix. Dataset-level assessment results

See Tables A.1 and A.2.

Table A.1 (*continued*).

Dataset name	K-mean-Euc	K-med-Euc	K-mean-shape	K-mean-DTW	C-mean-Euc	D-Peaks-Euc	D-Peaks-DTW	Agglo-Euc
GunPoint	−0.01	−0.01	−0.01	−0.01	−0.01	−0.01	−0.01	−0.01
GunPointAgeSpan	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GunPointMaleVersusFemale	0.23	0.23	0.00	0.23	0.23	0.23	0.23	0.23
GunPointOldVersusYoung	0.24	0.24	0.00	0.24	0.24	0.24	0.24	0.24
Ham	0.05	0.03	0.05	0.03	0.04	0.00	0.00	0.06
HandOutlines	0.29	0.28	0.32	0.04	0.29	0.01	0.00	0.39
Haptics	0.06	0.06	0.06	0.06	0.06	0.08	0.04	0.06
Herring	0.00	0.00	0.00	0.00	0.00	−0.01	0.03	0.02
HouseTwenty	0.11	0.11	0.11	0.18	0.12	0.16	−0.01	0.07
InlineSkate	0.01	0.01	0.04	0.04	0.01	0.01	0.02	0.01
InsectEPGRegularTrain	1.00	1.00	0.00	1.00	0.96	1.00	1.00	1.00
InsectEPGSmallTrain	0.91	0.91	0.00	1.00	1.00	1.00	1.00	1.00
InsectWingbeatSound	0.34	0.33	0.17	0.25	0.14	0.33	0.18	0.33
ItalyPowerDemand	0.00	0.35	0.01	0.00	0.00	0.54	0.17	0.00
LargeKitchenAppliances	0.02	0.02	0.01	0.03	0.02	0.01	0.06	0.02
Lightning7	0.26	0.22	0.35	0.20	0.15	0.23	0.18	0.30
Mallat	0.77	0.72	0.70	0.80	0.95	0.58	0.43	0.84
Meat	0.62	0.62	0.46	0.55	0.49	0.82	0.45	0.44
MedicalImages	0.05	0.04	0.08	0.05	0.05	0.04	−0.04	0.04
MelbournePedestrian	0.44	0.45	0.10	0.41	0.43	0.41	0.24	0.47
MiddlePhalanxOutlineAgeGroup	0.35	0.34	0.39	0.42	0.42	0.01	−0.03	0.43
MiddlePhalanxOutlineCorrect	0.00	0.00	0.00	−0.01	0.00	−0.02	−0.02	−0.01
MiddlePhalanxTW	0.37	0.37	0.46	0.58	0.44	−0.01	0.11	0.37
MixedShapesRegularTrain	0.44	0.30	0.44	0.47	0.38	0.38	0.13	0.55
MixedShapesSmallTrain	0.46	0.40	0.48	0.53	0.41	0.40	0.52	0.55
MoteStrain	0.39	0.36	0.61	0.42	0.45	0.55	0.00	0.38
NonInvasiveFetalECGThorax1	0.43	0.38	0.33	0.35	0.15	0.12	0.08	0.47
NonInvasiveFetalECGThorax2	0.50	0.45	0.46	0.49	0.19	0.22	0.17	0.54
OliveOil	0.51	0.40	0.49	0.36	0.70	0.23	0.15	0.63
OSULeaf	0.14	0.12	0.24	0.13	0.05	0.07	0.01	0.18
PhalangesOutlinesCorrect	0.01	0.01	0.01	0.00	0.01	0.01	0.00	0.00
Phoneme	0.02	0.01	0.04	0.01	0.00	0.00	0.01	0.00
PigAirwayPressure	0.05	0.04	0.01	0.06	0.06	0.04	0.05	0.05
PigArtPressure	0.16	0.14	0.00	0.14	0.09	0.15	0.11	0.19
PigCVP	0.07	0.07	0.00	0.09	0.08	0.05	0.04	0.08
Plane	0.70	0.63	0.74	0.80	0.86	0.83	1.00	0.80
PowerCons	0.73	0.61	0.05	0.66	0.75	0.15	0.00	0.86
ProximalPhalanxOutlineAgeGroup	0.42	0.43	0.50	0.57	0.51	0.35	0.02	0.52
ProximalPhalanxOutlineCorrect	0.07	0.06	0.07	0.05	0.07	0.06	0.11	0.05
ProximalPhalanxTW	0.40	0.40	0.44	0.32	0.38	0.25	0.33	0.42
RefrigerationDevices	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00
Rock	0.22	0.19	0.06	0.23	0.23	−0.01	0.23	0.30
ScreenType	0.02	0.01	0.01	0.01	0.03	0.00	0.00	0.02
SemgHandGenderCh2	0.00	0.00	0.13	0.00	−0.01	0.01	−0.01	0.00
SemgHandMovementCh2	0.14	0.14	0.05	0.16	0.14	0.01	0.00	0.13
SemgHandSubjectCh2	0.08	0.08	0.12	0.10	0.07	0.03	0.00	0.10
ShapeletSim	0.00	0.01	0.46	0.00	0.00	0.00	0.00	0.00
ShapesAll	0.36	0.31	0.36	0.35	0.06	0.12	0.12	0.37
SmallKitchenAppliances	0.00	0.03	0.00	0.07	0.00	0.00	0.00	0.00
SmoothSubspace	0.44	0.29	0.18	0.43	0.43	0.35	0.03	0.50
SonyAIBORobotSurface1	0.34	0.23	0.46	0.71	0.53	0.03	0.00	0.41
SonyAIBORobotSurface2	0.32	0.21	0.18	0.30	0.32	−0.03	−0.02	0.26
StarLightCurves	0.52	0.35	0.53	0.53	0.52	0.54	0.68	0.51
Strawberry	−0.02	0.01	−0.02	−0.01	0.00	−0.04	0.08	−0.05
SwedishLeaf	0.30	0.28	0.32	0.15	0.27	0.09	0.04	0.30
Symbols	0.64	0.58	0.71	0.65	0.67	0.38	0.82	0.68
SyntheticControl	0.59	0.34	0.60	0.64	0.52	0.26	0.31	0.61
ToeSegmentation1	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.02
ToeSegmentation2	0.00	0.00	0.27	0.00	0.00	0.02	−0.01	0.05
Trace	0.34	0.35	0.32	0.41	0.34	0.34	0.66	0.33
TwoLeadECG	0.00	0.00	0.08	0.02	0.00	0.00	0.03	0.00
TwoPatterns	0.02	0.02	0.21	0.07	0.02	0.08	0.29	0.02
UMD	0.15	0.13	0.14	0.15	0.15	0.12	0.21	0.14
UWaveGestureLibraryAll	0.55	0.50	0.62	0.52	0.17	0.54	0.25	0.59
UWaveGestureLibraryX	0.34	0.32	0.30	0.39	0.32	0.40	0.49	0.41
UWaveGestureLibraryY	0.33	0.30	0.24	0.35	0.30	0.23	0.26	0.34
UWaveGestureLibraryZ	0.31	0.29	0.34	0.34	0.31	0.31	0.28	0.29
Wine	0.00	0.00	−0.01	−0.01	−0.01	0.00	−0.01	−0.01

(*continued on next page*)

Table A.1 (*continued*).

Dataset name	K-mean-Euc	K-med-Euc	K-mean-shape	K-mean-DTW	C-mean-Euc	D-Peaks-Euc	D-Peaks-DTW	Agglo-Euc
WordSynonyms	0.16	0.14	0.19	0.23	0.10	0.14	0.18	0.17
Worms	0.02	0.00	0.05	0.02	0.01	0.00	0.00	0.07
WormsTwoClass	0.00	0.00	0.00	0.00	0.00	0.00	0.00	−0.01
Yoga	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.2
Pairwise spread of ARI scores between clustering methods.

Clustering method	Agglo-merative (Euc)	K-means (DTW)	K-means (Euc)	C-means (Euc)	K-med (Euc)	K-means (shape)	Density peaks (Euc)	Density Peaks (DTW)
Agglomerative (Euclidean)	–	0.020	0.004	0.011	0.011	0.050	0.043	0.054
K-means (DTW)	–	–	0.016	0.025	0.017	0.041	0.043	0.052
K-means (Euclidean)	–	–	–	0.010	0.005	0.043	0.036	0.045
C-means (Euclidean)	–	–	–	–	0.011	0.053	0.038	0.043
K-medoids (Euclidean)	–	–	–	–	–	0.042	0.021	0.032
K-means (shape-based)	–	–	–	–	–	–	0.060	0.067
Density Peaks (Euclidean)	–	–	–	–	–	–	–	0.021
Density Peaks (DTW)	–	–	–	–	–	–	–	–

References

Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering – A decade review. *Information Systems*, *53*, 16–38. http://dx.doi.org/10.1016/j.is.2015.04.007.

Ali, M., Alqahtani, A., Jones, M. W., & Xie, X. (2019). Clustering and classification for time series data in visual analytics: A survey. *IEEE Access*, *7*, 181314–181338. http://dx.doi.org/10.1109/ACCESS.2019.2958551.

Begum, N., Ulanova, L., Wang, J., & Keogh, E. (2015). Accelerating dynamic time warping clustering with a novel admissible pruning strategy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 49–58). http://dx.doi.org/10.1145/2783258.2783286.

Bende-Michl, U., Verburg, K., & Cresswell, H. P. (2013). High-frequency nutrient monitoring to infer seasonal patterns in catchment source availability, mobilisation and delivery. *Environmental Monitoring and Assessment*, *185*(11), 9191–9219. http://dx.doi.org/10.1007/s10661-013-3246-8.

Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer, http://dx.doi.org/10.1007/978-1-4757-0450-1.

Bezdek, J. C., & Pal, N. R. (1998). Some new indexes of cluster validity. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *28*(3), 301–315. http://dx.doi.org/10.1109/3477.678624.

Bholowalia, P., & Kumar, A. (2014). EBK-means: A clustering technique based on elbow method and K-means in WSN. *International Journal of Computer Applications*, *105*, 17–24.

Caliński, T., & J.A., H. (1974). A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, *3*, 1–27. http://dx.doi.org/10.1080/03610927408827101.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., & Keogh, E. (2018). The UCR time series archive. arXiv:1810.07758.

Dau, H. A., Keogh, E., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., & Hexagon-ML (2018). The UCR time series classification archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *1*(2), 224–227. http://dx.doi.org/10.1109/TPAMI.1979.4766909.

Ding, H., Trajcevski, G., Scheuermann, P., & Keogh, E. (2010). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, *26*, 275–309. http://dx.doi.org/10.1007/s10618-012-0250-5.

Du, M., Ding, S., Xue, Y., & Shi, Z. (2019). A novel density peaks clustering with sensitivity of local density and density-adaptive metric. *Knowledge and Information Systems*, *59*(2), 285–309. http://dx.doi.org/10.1007/s10115-018-1189-7.

Dupas, R., Tavenard, R., Fovet, O., Gilliet, N., Grimaldi, C., & Gascuel-Odoux, C. (2015). Identifying seasonal patterns of phosphorus storm dynamics with dynamic time warping. *Water Resources Research*, *51*(11), 8868–8882. http://dx.doi.org/10.1002/2015WR017338.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (pp. 226–231).

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data* (pp. 419–429). http://dx.doi.org/10.1145/191839.191925.

Flanagan, K., Fallon, E., Connolly, P., & Awad, A. (2017). Network anomaly detection in time series using distance based outlier detection with cluster density analysis. In *Proceedings of the 2017 Internet Technologies and Applications* (pp. 116–121).

Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, *78*(383), 553–569. http://dx.doi.org/10.1080/01621459.1983.10478008.

Fränti, P., & Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. *Applied Intelligence*, *48*(12), 4743–4759. http://dx.doi.org/10.1007/s10489-018-1238-7.

Großwendt, A., Röglin, H., & Schmidt, M. (2019). Analysis of Ward's method. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 2939–2957).

Gupta, K., & Chatterjee, N. (2018). Financial time series clustering. In *Information and Communication Technology for Intelligent Systems (ICTIS 2017), vol. 2* (pp. 146–156).

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Iorio, C., Frasso, G., D'Ambrosio, A., & Siciliano, R. (2018). A P-spline based clustering approach for portfolio selection. *Expert Systems with Applications*, *95*, 88–103. http://dx.doi.org/10.1016/j.eswa.2017.11.031.

Javed, A. (2019). Time series clustering benchmark. https://github.com/ali-javed/clusteringBenchmark.

Javed, A., Hamshaw, S. D., Rizzo, D. M., & Lee, B. S. (2019). Analysis of hydrological and suspended sediment events from Mad River watershed using multivariate time series clustering. arXiv:1911.12466.

Johnpaul, C., Prasad, M. V., Nickolas, S., & Gangadharan, G. (2020). Trendlets: A novel probabilistic representational structures for clustering the time series data. *Expert Systems with Applications*, *145*, Article 113119. http://dx.doi.org/10.1016/j.eswa.2019.113119.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. John Wiley, http://dx.doi.org/10.1002/9780470316801.

Kaufman, L., & Rousseeuw, P. (2008). Origins and extensions of the K-means algorithm in cluster analysis. *Journal Électronique d'Histoire des Probabilités et de la Statistique [electronic only]*, *4*, 2–18.

Keogh, E. J., & Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery*, *7*(4), 349–371. http://dx.doi.org/10.1023/A:1024988512476.

Li, Z., & de Rijke, M. (2017). The impact of linkage methods in hierarchical clustering for active learning to rank. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 941–944). http://dx.doi.org/10.1145/3077136.3080684.

Liao, T. W. (2005). Clustering of time series data: A survey. *Pattern Recognition*, *38*(11), 1857–1874. http://dx.doi.org/10.1016/j.patcog.2005.01.025.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1* (pp. 281–297). https://projecteuclid.org/euclid.bsmsp/1200512992.

Mather, A. L., & Johnson, R. L. (2015). Event-based prediction of stream turbidity using a combined cluster analysis and classification tree approach. *Journal of Hydrology*, *530*, 751–761. http://dx.doi.org/10.1016/j.jhydrol.2015.10.032.

Maulik, U., & Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(12), 1650–1654. http://dx.doi.org/10.1109/TPAMI.2002.1114856.

Mechelen, I. V., Boulesteix, A.-L., Dangl, R., Dean, N., Guyon, I., Hennig, C., Leisch, F., & Steinley, D. (2018). Benchmarking in cluster analysis: A white paper. arXiv:1809.10496.

Minaudo, C., Dupas, R., Gascuel-Odoux, C., Fovet, O., Mellander, P.-E., Jordan, P., Shore, M., & Moatar, F. (2017). Nonlinear empirical modeling to estimate phosphorus exports using continuous records of turbidity and discharge. *Water Resources Research*, *53*, 7590–7606. http://dx.doi.org/10.1002/2017wr020590.

Mohamad, I., & Usman, D. (2013). Standardization and its effects on K-means clustering algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, *6*, 3299–3303. http://dx.doi.org/10.19026/rjaset.6.3638.

Paparrizos, J., & Gravano, L. (2016). K-shape: Efficient and accurate clustering of time series. *SIGMOD Record*, *45*(1), 69–76. http://dx.doi.org/10.1145/2949741.2949758.

Paparrizos, J., & Gravano, L. (2017). Fast and accurate time-series clustering. *ACM Transactions on Database Systems*, *42*(2), 8:1–8:49. http://dx.doi.org/10.1145/3044711.

Patil, C., & Baidari, I. (2019). Estimating the optimal number of clusters k in a dataset using data depth. *Data Science and Engineering*, *4*, 132–140.

Pirim, H., Ekşioğlu, B., Perkins, A. D., & Yüceer, C. (2012). Clustering of high throughput gene expression data. *Computers & Operations Research*, *39*, 3046–3061.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., & Keogh, E. (2012). Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 262–270). http://dx.doi.org/10.1145/2339530.2339576.

Roddick, J. F., & Spiliopoulou, M. (2002). A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, *14*(4), 750–767. http://dx.doi.org/10.1109/TKDE.2002.1019212.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, *344*(6191), 1492–1496. http://dx.doi.org/10.1126/science.1242072.

Romano, S., Vinh, N. X., Bailey, J., & Verspoor, K. (2016). Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research (JMLR)*, *17*(1), 4635–4666.

Rosenberg, A., & Hirschberg, J. (2007). V-Measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 410–420).

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*(1), 53–65. http://dx.doi.org/10.1016/0377-0427(87)90125-7.

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

Santos, J. M., & Embrechts, M. (2009). On the use of the Adjusted Rand Index as a metric for evaluating supervised classification. In *Proceedings of the 19th International Conference on Artificial Neural Networks: Part II* (pp. 175–184). http://dx.doi.org/10.1007/978-3-642-04277-5_18.

Souto, M. d., Costa, I., Araujo, D., Ludermir, T., & Schliep, A. (2008). Clustering cancer gene expression data: A comparative study. *BMC Bioinformatics*, *9:497*, http://dx.doi.org/10.1186/1471-2105-9-497.

Subbalakshmi, C., Krishna, G. R., Rao, S. K. M., & Rao, P. V. (2015). A method to find optimum number of clusters based on fuzzy Silhouette on dynamic data set. *Procedia Computer Science*, *46*, 346–353. http://dx.doi.org/10.1016/j.procs.2015.02.030.

Wu, X., & Kumar, V. (2009). *The Top Ten Algorithms in Data Mining*. Chapman & Hall/CRC.