2021

Cluster Analysis of Time Series Data with Application to Hydrological Events and Serious Illness Conversations

Ali Javed
University of Vermont

Cluster Analysis of Time Series Data with Application to Hydrological Events and Serious Illness Conversations

A Dissertation Presented

by

Ali Javed

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
Specializing in Computer Science

August, 2021

Defense Date: June 18th, 2021
Dissertation Examination Committee:

Donna M. Rizzo, Ph.D., Advisor
Byung Suk Lee, Ph.D., Advisor
Robert Gramling, M.D., D.Sc., Chairperson
Scott Hamshaw, Ph.D.
Nicholas Cheney, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of the Graduate College

Abstract

Cluster analysis explores the underlying structure of data and organizes it into groups (i.e., clusters) such that observations within the same group are more similar than those in different groups. Quantifying the "similarity" between observations, choosing the optimal number of clusters, and interpreting the results all require careful consideration of the research question at hand, the model parameters, the amount of data and their attributes. In this dissertation, the first manuscript explores the impact of design choices and the variability in clustering performance on different datasets. This is demonstrated through a benchmark study consisting of 128 datasets from the University of California, Riverside time series classification archive. Next, a multivariate event time series clustering approach is applied to hydrological storm events in watershed science. Specifically, river discharge and suspended sediment data from six watersheds in the Vermont are clustered, and yield four types of hydrological water quality events to help inform conservation and management efforts. In a second application, a novel and computationally efficient clustering algorithm called SOMTimeS (Self-organizing Map for Time Series) is designed for large time series analysis using dynamic time warping (DTW). The algorithm scales linearly with increasing data, making SOMTimeS, to the best of our knowledge, the fastest DTW-based clustering algorithm to date. For proof of concept, it is applied to conversational features from a Palliative Care Communication Research Initiative study with the goal of understanding and motivating high quality communication in serious illness health care settings.

Citations

Material from this dissertation has been accepted for publication in Machine Learning with Applications on July, 6, 2020. in the following form:
Javed, A., Lee B.S., Rizzo, D.M.. (2020). A benchmark study on time series clustering. Machine Learning with Applications, 1, 100001.

Material from this dissertation has been accepted for publication in Journal of Hydrology on November, 23, 2020. in the following form:
Javed, A., Hamshaw, S.D., Lee B.S., Rizzo, D.M.. (2020). Multivariate event time series analysis using hydrological and suspended sediment data. Journal of Hydrology, 593, 125802.

This dissertation is dedicated to my loving parents, Khalid and Farzana Javed, who have been a constant source of support and encouragement for me.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The recent explosion in time series data is due to the increase in sensor development and other data generating devices (CRS, 2020; Evans, 2011) as well as their reduced cost (UNEP, 2021). With this increase in time series data, the need for methods capable of clustering and classifying the data abound in a variety of disciplines. Examples include hydrological storm event analysis (Minaudo et al., 2017; Dupas et al., 2015; Mather and Johnson, 2015; Bende-Michl et al., 2013), conversations (Ross et al., 2020), financial portfolio building (Iorio et al., 2018), enhanced index tracking (Gupta and Chatterjee, 2018), personalized drug design (Pirim et al., 2012), cancer sub-type identification (Souto et al., 2008), and anomaly detection (Flanagan et al., 2017), among others. Ultimately, how these data are used, and whether we can successfully tackle the well-known challenges of structure and scale will depend on who can translate this data and what they do with it.

Several aspects of data, available resources, and study objective affect the choice of methodology in a cluster analysis. For instance, temporality of data affects the quantification of similarity (Begum et al., 2015; Liao, 2005), the attributes of data (e.g., outliers and number of features) affects the choice of clustering algorithm (Jin and Han, 2010), the amount of data (e.g., number and length of time series) in relation to available resources (i.e., computational power, available memory and time

constraint) affects choice of data preprocessing routines (e.g., sampling and reducing sequence length), and the study objective often guides the feature selection. For clustering the methodology needs to be tailored to the application domain of the studied analysis without labeled data to guide selection.

There are three types of time series clustering algorithms (Jin and Han, 2010) — 1) raw-data-based, 2) feature-based, and 3) model-based. This dissertation focuses on the first type (i.e., raw-data-based). Raw-data-based approaches differentiate themselves from the other two by preserving the temporal order of variables within the input observation (Jin and Han, 2010). Within the first type, I focus on clustering algorithm, distance measure and application domain; the cluster analysis is tailored to the study objective.

1.1 Study Problems

This dissertation is divided into three parts. In the first part a benchmark study for time series clustering algorithms is presented. In the second and third part, selected cluster analysis methods are applied to two disparate fields of research – the clustering of hydrological storm events in watershed science to inform watershed conservation and management efforts and the clustering of conversations in serious illness to understand and incentivize high quality communication. Each part tailors several aspects of time series clustering to the research problem and the study objective.

The benchmark study in the first part is based on the fact that, given the exploratory nature of cluster analysis, different choices of methods can result in different outcomes. Consequently, clustering methods show high variability in performance (measured in terms of the ability to regenerate specific ground truths) across datasets. Lack of availability of dataset-level performance metrics for different clustering algorithms forces researchers to repeat benchmarking studies for algorithm

selection. The benchmark used in this work comprises all 128 datasets available in the University of California Riverside (UCR) archive. It 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). A phased evaluation approach was designed for summarizing dataset-level assessment metrics and discussing the results.

In the second part of this dissertation, hydrological storm events are modeled as multivariate time series and clustered using a multivariate event time series (METS) clustering method. Hydrological storm events are primary drivers for transporting water quality constituents such as suspended sediments and nutrients (Dupas et al., 2015; Sherriff et al., 2016). Analyzing the concentration (C) of these water quality constituents in response to river discharge (Q), particularly when monitored at high temporal resolution during a hydrological event, helps to characterize the dynamics and flux of such constituents (Aguilera and Melack, 2018; Burns et al., 2019; Williams et al., 2018; Malutta et al., 2020). The proposed METS approach is the first to incorporate temporal details of the C-Q relationship for the purpose of identifying event types. The METS clustering was applied to river discharge and suspended sediment data (acquired through turbidity-based monitoring) from six watersheds in the Lake Champlain Basin located in the northeastern United States, and results in identifying four common types of hydrological water quality events.

In the third part, serious illness conversations are modeled as story arcs (i.e., narrative times) and are clustered using a self-organizing maps for time series (SOMTimeS) clustering method. SOMTimeS combines Kohonen self-organizing maps (Kohonen et al., 2001) and dynamic time warping (DTW) (Sakoe and Chiba, 1978) in a computationally efficient clustering algorithm. A framework of conversational storytelling is a very useful conceptual foundation for studying communication in serious illness. Conversational storytelling is a practical and

effective way humans find and share meaning with one another. In the study, SOMTimeS is used to discover fundamental shapes of serious illness conversational stories. SOMTimeS was applied to 171 conversations obtained from a Palliative Care Communication Research Initiative (PCCRI) study, and resulted in identifying two fundamental shapes of conversational stories in serious illness.

1.2 Dissertation Outline

This dissertation is organized in a journal format according to the University guidelines. It comprises three parts. The first part (Chapter 2) presents a benchmark of time series clustering methods. The second part (Chapters 3) presents METS clustering approach and its application to hydrological storm events. The third part (Chapter 4) presents SOMTimeS clustering method and its application to serious illness conversations. Finally, Chapter 5 concludes the dissertation.

Chapter 2

A Benchmark Study on Time Series Clustering

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.

2.1 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 and 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 and 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 (Javed et al., 2020; Minaudo et al., 2017; Dupas et al., 2015; Mather and Johnson, 2015; Bende-Michl et al., 2013) 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; Paparrizos and Gravano, 2016, 2017; Du et al., 2019; Begum et al., 2015). 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 (Paparrizos and Gravano, 2016, 2017; Begum et al., 2015) of a given time series data.

The University of California (UCR) time series archive (Dau et al., 2018b) 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 and Chiba, 1978).

Despite extensive use of the archive in creating, validating and evaluating some of the most recently popular time series clustering algorithms (Paparrizos and Gravano, 2016, 2017; Begum et al., 2015), 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 and Gravano, 2017); and when 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 et al. (2018b), 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 et al., 2018b; Keogh and 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., Bholowalia and Kumar, 2014; Patil and Baidari, 2019; Subbalakshmi et al., 2015; Bezdek and Pal, 1998), 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 2.2 discusses related work. Section 2.3 describes the benchmark methods. Section 2.4 presents the benchmark test results. Section 2.5 highlights the limitations and related opportunities of the benchmark. Section 2.6 concludes the paper.

2.2 Related work

Benchmarking, in general, has been recognized as an important step in advancing the knowledge of both supervised and unsupervised learning (Keogh and Kasetty, 2003; Mechelen et al., 2018; Ding et al., 2010; Fränti and Sieranoja, 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., Liao, 2005; Ali et al., 2019; Roddick and Spiliopoulou, 2002), or quantitatively using an empirical approach (e.g., Paparrizos and Gravano, 2016, 2017; Begum et al., 2015). ts 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., Paparrizos and Gravano, 2016, 2017; Begum et al., 2015). 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.

2.3 Benchmark Methods

The benchmark methods comprise clustering methods (Section 2.3.1) and evaluation methods (Section 2.3.2).

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 2.1). 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.

Table 2.1: Eight benchmark clustering methods. [1] (Paparrizos and Gravano, 2016), [2] (Sakoe and 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

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 and Rousseeuw, 1990), and Fuzzy C-means (Bezdek, 1981), are selected based on their popularity (Ali et al., 2019) and known accuracy

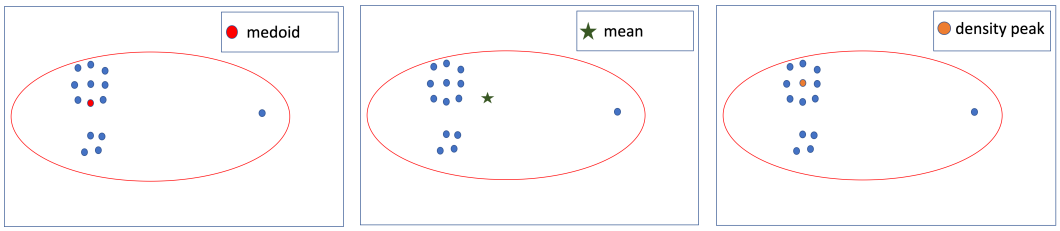


Figure 2.1: Different types of centroids: (a) medoid in K-medoids, (b) centroid in K-means, and (c) density peak in Density Peaks.

for time series data clustering (Paparrizos and Gravano, 2017). Note K-means with shape-based distance is K-shape (Paparrizos and 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 Figure 2.1a and Figure 2.1b).

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 and 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 2.1).

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 Figure 2.1c) 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 and Laio, 2014).

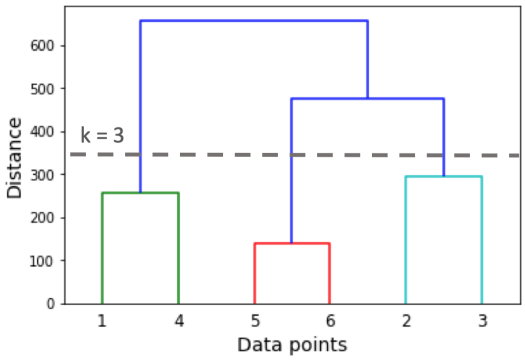


Figure 2.2: Agglomerative clustering.

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 Figure 2.2). 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 and de Rijke, 2017).

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

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). Equation 2.1 shows how the Euclidean distance $d(T1, T2)$ is calculated between two time series $T1 = (T1_1, T1_2, ..., T1_n)$ and $T2 = (T2_1, T2_2, ..., T2_n)$.

$$d(T1, T2) = \sqrt{\sum_{i}^{n} (T1_i - T2_i)^2} \tag{2.1}$$

Dynamic time warping

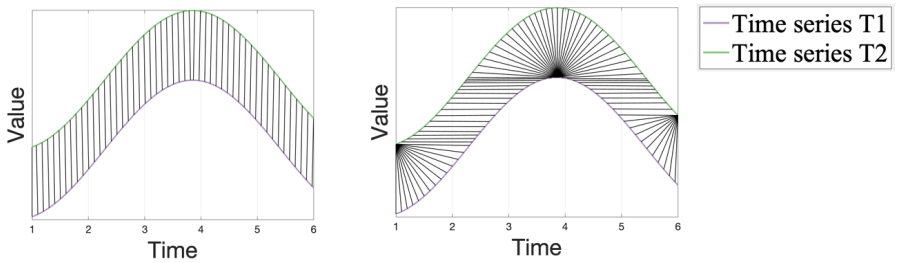


Figure 2.3: Alignment between two times series for calculating distance.

Dynamic time warping (DTW) is a mapping of points between a pair of time series, $T1$ and $T2$ (see Figure 2.3) designed to minimize the pairwise Euclidean distance. It is becoming recognized as one of the most accurate similarity measures for time series data (Paparrizos and Gravano, 2017; Rakthanmanon et al., 2012; Johnpaul et al., 2020). 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 and 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 and 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 and Gravano, 2016):

$$T' = \frac{T - \mu}{\sigma} \tag{2.2}$$

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

2.3.2 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.

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 and Arabie, 1985), Adjusted Rand Index (ARI) (Santos and Embrechts, 2009), Adjusted Mutual Information (AMI) (Romano et al., 2016), Fowlkes Mallows index (FMS) (Fowlkes and Mallows, 1983), Homogeneity (Rosenberg and Hirschberg, 2007), and Completeness (Rosenberg and 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 and Bouldin, 1979), Calinski- Harabasz index (Calinski and JA, 1974), the I-index (Maulik and 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 (Figure 2.4f) for prior UCR archive studies (e.g., Paparrizos and Gravano, 2016, 2017; Begum et al., 2015), we find the adjusted measures more suitable for clustering because they are independent of the number of clusters. As demonstrated in Figure 2.4, the accuracy scores resulting from random cluster assignment are consistently low as the number of clusters varies for the two adjusted measures (Figures 2.4a and 2.4b), 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.

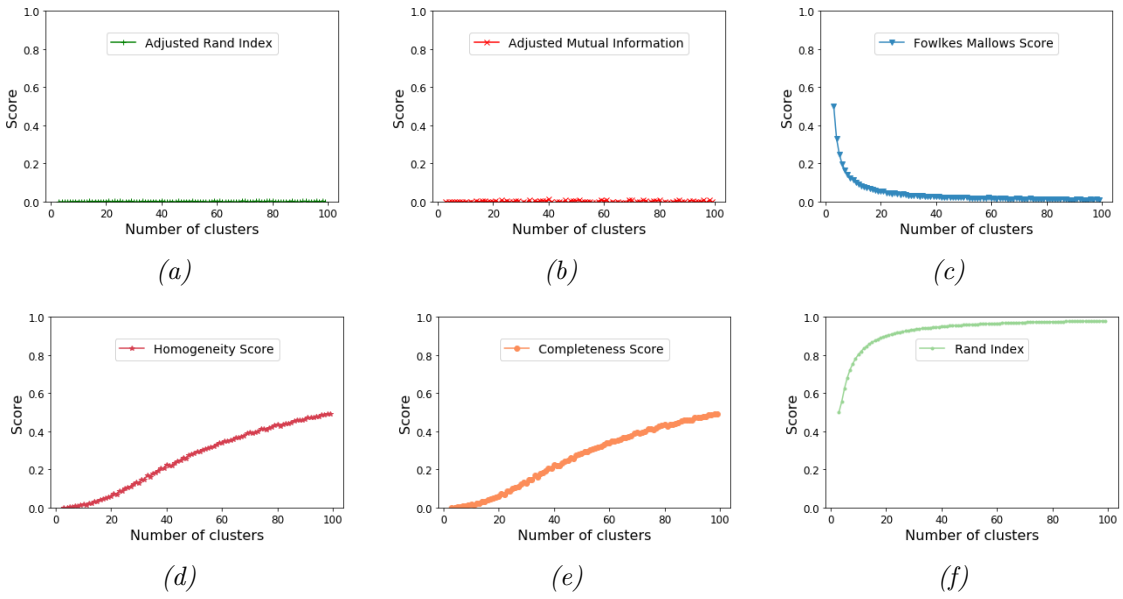


Figure 2.4: Accuracy scores resulting from randomly assigning 1000 data points to a varying number of clusters.

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 i^{th} cluster of X and the j^{th} cluster of Y, the Adjusted Rand Index is calculated as shown in Equation 2.3.

$$\text{Adjusted Rand Index} = \frac{\sum_i^j \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{2.3}$$

where a_i is the sum of the i^{th} row and b_j is the sum of the j^{th} 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 Equation 2.4).

$$\text{Spread} = \frac{\sum_{i=1}^{n} (A1_i - A2_i)^2}{n} \tag{2.4}$$

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.

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. 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., 3,136 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 2.3.2) 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.

2.4 Benchmark Test Results

This section provides the results of dataset-level assessment (Section 2.4.1) and the phased evaluation (Section 2.4.2), and discusses the results (Section 2.4.3).

2.4.1 Dataset-level assessment

2.7 shows the Adjusted Rand Index (ARI) scores for all eight clustering methods on the 112 short-listed datasets (see Section 2.1) 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 2.1), the scores of each clustering method on each dataset tested for all the six external measures (see Section 2.3.2) are available at GitHub (Javed, 2019) along with the source codes.

2.4.2 Phased evaluation

Phase 1 - Ranked comparison of all methods Figure 2.5 shows the average ARI's for each of the eight clustering methods in decreasing order. In addition, Table 2.2 and Table 2.3 provide details, including the average and standard

deviation of the clustering scores resulting from the six external assessment metrics (see Section 2.3.2). Table 2.2 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.

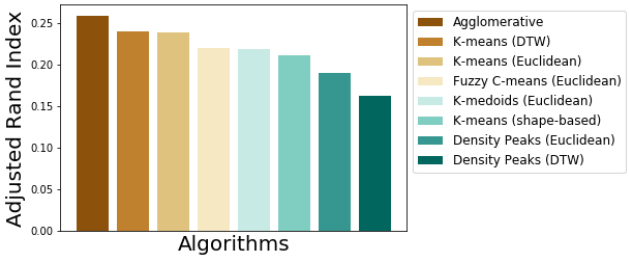


Figure 2.5: Average ARI for each clustering method in Phase 1.

Table 2.2: 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 2.3 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

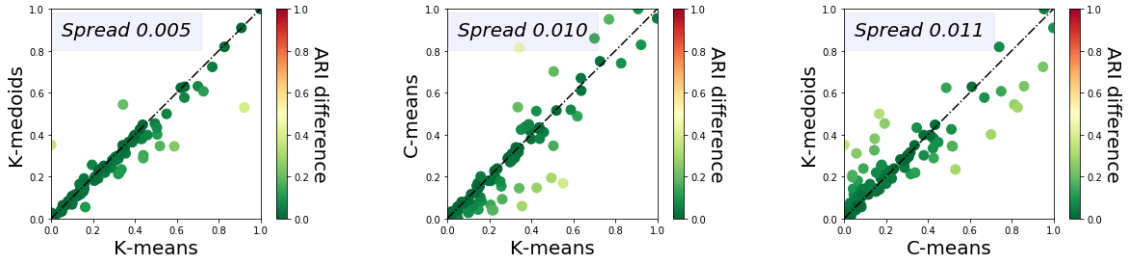
Table 2.3: Average and standard deviation of non-adjusted measures for each clustering method in Phase 1.

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

not so meaningful in this benchmark. For instance, for certain datasets such as GunPointAgeSpan, GunPoindMaleVersusFemale and GunPointOldVersusYoung (see 2.7), 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 Table 2.2 and Table 2.3 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 2.4). While K-means had a higher ARI score in almost twice as many datasets, differences in score values were minor, with a spread of only 0.005 against K-medoids (see Figure 2.6a) and only slightly



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

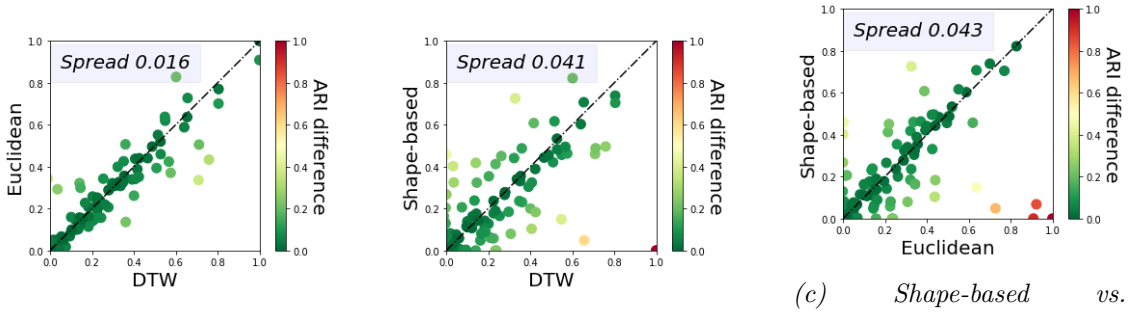
Figure 2.6: Spread of ARI scores between each pair of the three clustering algorithms with Euclidean distance in Phase 2.

Table 2.4: 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

larger (0.010) against Fuzzy C-means (see Figure 2.6b). 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 2.5). 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 Figure 2.7



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

Figure 2.7: Spread of ARI scores between each pair of distance measures in Phase 3

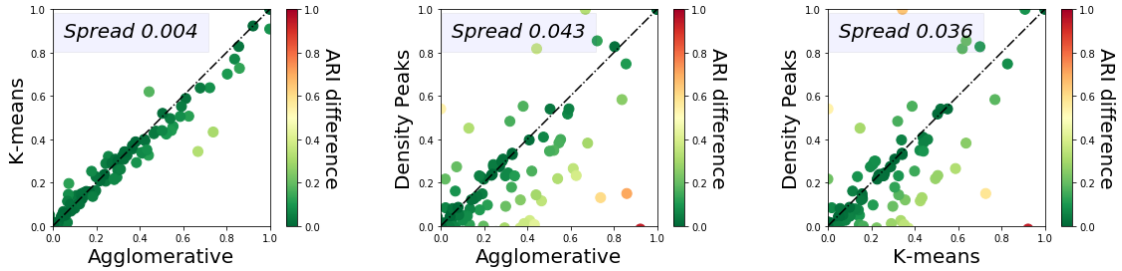
Table 2.5: Different distance measures for K-means (from Phase 2) in Phase 3.

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

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 2.6, 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 Figure 2.8a) is fairly small compared with the spread of either method with Density Peaks (see Figure 2.8b and Figure 2.8c).



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

Figure 2.8: Different algorithms with Euclidean distance measure in Phase 4.

Table 2.6: Different algorithms with Euclidean distance measure in Phase 4.

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

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 2.7) when Euclidean distance was used as the distance measure compared to a count of 31 with DTW. Figure 2.9 shows the spread of ARI scores between Euclidean distance and DTW to be 0.021.

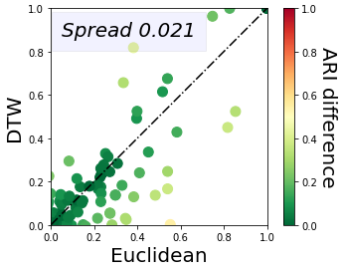


Figure 2.9: Euclidean vs. DTW for Density Peaks algorithm in Phase 5.

Table 2.7: Euclidean vs. DTW for Density Peaks algorithm in Phase 5.

Distance measure	Winning count
Euclidean	45
DTW	31

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 2.8) 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 Figure 2.10) was the largest (0.052) observed in the six phases.

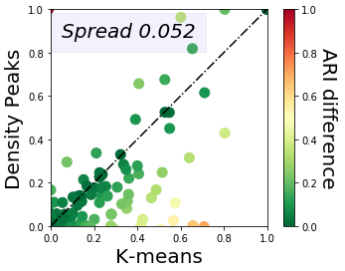


Figure 2.10: DTW in Density Peaks and K-means (selected in Phase 2) in Phase 6.

Table 2.8: DTW in Density Peaks and K-means (selected in Phase 2) in Phase 6.

Algorithm	Winning count
K-means	60
Density Peaks	24

2.4.3 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 Table 2.2 and Table 2.3 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 et al., 2018b). 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 2.4), 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 (Figure 2.6b) and K-medoids (Figure 2.6c), 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 2.5 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 and 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 (Figures 2.7b and 2.7c) 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.

Phase 4 - Comparison of clustering algorithms using Euclidean distance The very small spread in Figure 2.8a 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 Figure 2.9) in Density Peaks algorithm is relatively consistent with spread (0.016) between DTW and Euclidean in K-means algorithm (see Figure 2.7a). 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 2,133 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 Figure 2.10), which is consistent with the somewhat high spread 0.036 observed between the two methods (see Phase 4 with Euclidean distance, Figure 2.8c). 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 Figure 2.1).

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 2.2 and Figure 2.5), is higher than the variation observed across winning counts (Table 2.4 to Table 2.8). 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 and 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 Figure 2.11). 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 Figure 2.7c, Figure 2.8a and Figure 2.9) on the same dataset are more likely to provide three dissimilar clustering outputs, compared to those generated using

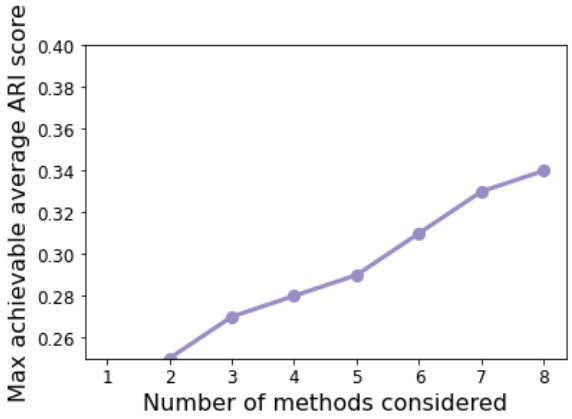


Figure 2.11: Maximum achievable average ARI score for progressively increasing number of methods (over time).

K-means (Euclidean), K-medoids (Euclidean), and Fuzzy C-means (Euclidean) (lower spread values in Figure 2.6).

2.5 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 et al., 2018a), 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.

2.6 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—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 2.4.3) 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 2.3.2 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 2.5).

Acknowledgements

This project was supported by the grant from the Barrett Foundation and Gund Institute for Environment through a Gund Barrett PhD 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.

Bibliography

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

Ali, M., Alqahtani, A., Jones, M. W., and Xie, X. (2019). Clustering and classification for time series data in visual analytics: A survey. *IEEE Access*, 7:181314–181338.

Begum, N., Ulanova, L., Wang, J., and 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*, pages 49–58.

Bende-Michl, U., Verburg, K., and 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.

Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer.

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

Bholowalia, P. and 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.

Calinski, T. and JA, H. (1974). A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3:1–27.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., and Keogh, E. (2018a). The UCR time series archive. aiXrv 1801.07758.

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., and Hexagon-ML (2018b). The UCR time series classification archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227.

Ding, H., Trajcevski, G., Scheuermann, P., and Keogh, E. (2010). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26:275–309.

Du, M., Ding, S., Xue, Y., and 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.

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

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

Faloutsos, C., Ranganathan, M., and Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, pages 419–429.

Flanagan, K., Fallon, E., Connolly, P., and 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*, pages 116–121.

Fowlkes, E. B. and Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569.

Fränti, P. and Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. *Applied Intelligence*, 48(12):4743–4759.

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

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

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

Iorio, C., Frasso, G., D'Ambrosio, A., and Siciliano, R. (2018). A P-spline based clustering approach for portfolio selection. *Expert Systems with Applications*, 95:88 – 103.

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

Javed, A., Hamshaw, S. D., Lee, B. S., and Rizzo, D. M. (2020). Multivariate event time series analysis using hydrological and suspended sediment data. *Journal of Hydrology*, page 125802.

Johnpaul, C., Prasad, M. V., Nickolas, S., and Gangadharan, G. (2020). Trendlets: A novel probabilistic representational structures for clustering the time series data. *Expert Systems with Applications*, 145:113119.

Kaufman, L. and Rousseeuw, P. (2008). Origins and extensions of the K-means

algorithm in cluster analysis. *Journal Electronique d'Histoire des Probabilites et de la Statistique [electronic only]*, 4:2–18.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley.

Keogh, E. J. and 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.

Li, Z. and 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*, pages 941–944.

Liao, T. W. (2005). Clustering of time series data: A survey. *Pattern Recognition*, 38(11):1857 – 1874.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297.

Mather, A. L. and 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.

Maulik, U. and Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654.

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

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

Mohamad, I. and Usman, D. (2013). Standardization and its effects on K-means clustering algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, 6:3299–3303.

Paparrizos, J. and Gravano, L. (2016). K-shape: Efficient and accurate clustering of time series. *SIGMOD Record*, 45(1):69–76.

Paparrizos, J. and Gravano, L. (2017). Fast and accurate time-series clustering. *ACM*

Transactions on Database Systems, 42(2):8:1–8:49.

Patil, C. and 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., and Yüceer, C. (2012). Clustering of high throughput gene expression data. *Computers and Operations Research*, 39:3046–3061.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and 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*, pages 262–f–270.

Roddick, J. F. and Spiliopoulou, M. (2002). A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767.

Rodriguez, A. and Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496.

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

Rosenberg, A. and 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*, pages 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.

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

Santos, J. M. and 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*, pages 175–184.

Souto, M. d., Costa, I., Araujo, D., Ludermir, T., and Schliep, A. (2008). Clustering cancer gene expression data: A comparative study. *BMC Bioinformatics*, 9:497.

Subbalakshmi, C., Krishna, G. R., Rao, S. K. M., and 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.

Wu, X. and Kumar, V. (2009). *The Top Ten Algorithms in Data Mining*. Chapman & Hall/CRC, 1st edition.

2.7 Dataset-Level Assessment Results

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

Continued on next page

Table A.1 – continued from previous page

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
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

Continued on next page

Table A.1 – continued from previous page

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
Fungi	0.64	0.63	0.15	0.55	0.61	0.85	0.52	0.72
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

Continued on next page

Table A.1 – continued from previous page

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
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

Continued on next page

Table A.1 – continued from previous page

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
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
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

Continued on next page

Table A.1 – continued from previous page

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
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	Agglomerative (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)	-	-	-	-	-	-	-	-

Chapter 3

Multivariate Event Time Series Analysis using Hydrological and Suspended Sediment Data

ABSTRACT

Hydrological storm events are a primary driver for transporting water quality constituents such as suspended sediments and nutrients. Analyzing the concentration (C) of these water quality constituents in response to river discharge (Q), particularly when monitored at high temporal resolution during a hydrological event, helps to characterize the dynamics and flux of such constituents. A conventional approach to storm event analysis is to reduce C-Q time series to two-dimensional (2-D) hysteresis loops and analyze these 2-D patterns. While informative, this hysteresis loop approach has limitations because projecting the C-Q time series onto a 2-D plane obscures detail (e.g., temporal variation) associated with the C-Q relationships. In this paper, we address this limitation using a multivariate event time series (METS) clustering approach that is validated using synthetically generated event times series. The METS clustering is then applied to river discharge and suspended sediment data (acquired through turbidity-based monitoring) from six watersheds in the Lake Champlain Basin located in the northeastern United States, and results in identifying four common types of hydrological water quality events. Statistical analysis on the events partitioned by both methods (METS clustering and 2-D hysteresis classification) helped identify hydrometeorlogical features of common event types. In addition, the METS and hysteresis analysis were simultaneously applied to a regional Vermont dataset to highlight the complimentary nature of using them in tandem for hydrological event analysis.

3.1 Introduction

Characterizing the processes associated with rainfall-runoff events is an essential part of watershed research; and studying the dynamics that drive these processes (e.g., the timing and location of water quality constituent fluxes through the landscape) has many applications in the hydrological sciences. These include identifying sources of erosion present in a watershed (Sherriff et al., 2016), monitoring for shifts in watershed function (Burt et al., 2015), improving hydrological model forecasts (Ehret and Zehe, 2011), and informing watershed conservation and management efforts (Bende-Michl et al., 2013; Chen et al., 2017). Environmental managers and scientists often analyze hydrological data (e.g., suspended sediment concentration and streamflow) at an event scale — in this work, the period of storm-runoff resulting from a rainfall event – because this period is the primary mechanism for transporting many constituents of concern (Dupas et al., 2015; Sherriff et al., 2016). The timing of constituent delivery relative to stream discharge is complex and often exhibits a high degree of variability, especially when the monitoring frequency is high (Minaudo et al., 2017); and unsurprisingly, the relationship between multiple responses during a single event (e.g., discharge and water quality constituents) is often not linear (Onderka et al., 2012). However, despite the inherent complexity and dynamic behavior, the analysis of concentration-discharge (C-Q) relationships to infer mechanistic watershed processes at the event scale has a long tradition in hydrology, geomorphology and ecology (Aguilera and Melack, 2018; Burns et al., 2019; Williams et al., 2018; Malutta et al., 2020).

A fundamental feature of suspended sediment and solute transport in rivers is that the concentration of such constituents is often not in phase with the associated stream discharge, resulting in hysteresis being observed in the C-Q relationship. Williams (1989) was one of the first to use hysteresis patterns to study hydrological storm

events, identifying six classes of hydrological events and offering linkages between the hysteresis classes and watershed processes. While the study focused on suspended sediment concentration (SSC) data, these event classifications have been widely adopted in studies of both sediment and solutes, and continue to be used today to group storm events (e.g., Aguilera and Melack, 2018; Rose et al., 2018; Keesstra et al., 2019). An alternate to using 2D hysteresis patterns for categorization is to simplify the C-Q relationship into a scalar hysteresis index (Lloyd et al., 2016b). While both approaches are effective for inferring certain physical processes, each loses some information associated with the raw time series data, because both approaches "collapse" the time dimension, either by projecting the C-Q data onto a two-dimensional plane, or reducing the information into a scalar value (an index). Thus, temporal information associated with the original times series, such as the rate of change of a variable as well as aspects of its shape (e.g., linear, convex, concave), may be lost. With the increasing availability of high frequency sensors and associated data processing tools, it is now possible to leverage the temporal information embedded in multiple time series and fuse the data with complementary event analysis schemes such as hysteresis loop classification (Williams, 1989).

A few hydrological studies have used univariate time series (e.g., discharge) to quantify the similarity between storm events for forecasting purposes. Ehret and Zehe (2011) used manual feature extraction to propose a similarity measure for discharge time series that leverages hydrograph attributes such as the rising limb, peak and receding limb. Such manual feature extraction works well for hydrographs, but may not generalize to multivariate water quality time series. Ewen (2011) modified the minimal variance matching algorithm (Latecki et al., 2005) to quantify the similarity between two hydrographs. Presented with a hydrograph defined by a sequence of discharge measurements (called a "query sequence"), the method finds a target hydrograph that contains a sub-sequence most similar to the query sequence. Because

only a portion of the target sequence is matched (Latecki et al., 2005), similarity is not symmetric in both directions (i.e., $d(x, y)! = d(y, x)$) and, hence, may not be appropriate for use in clustering hydrological event data. Wendi et al. (2019) used recurrence quantification analysis and cross-recurrence plots to measure similarity between recurring hydrograph patterns. Recurrence quantification analysis is useful for large flood events (particularly those with multiple peaks); however, when the events are delineated, as is done in our work, the approach may not be appropriate. Regardless, none of the above classification methods were designed for analyzing events with multivariate time series.

Several studies have clustered storm events using event metrics and/or coefficients of best fit models. Dupas et al. (2015) used dynamic time warping (DTW) and K-means clustering to cluster re-scaled time series of phosphorus concentration. They manually select an ideal hydrograph and use the DTW algorithm to align each hydrograph in the dataset to the ideal hydrograph. Using these aligned hydrographs, the respective event phosphorus concentration graphs are then clustered to find dominant response patterns associated with physical processes occurring in the watershed. Bende-Michl et al. (2013) used high frequency data to build a database of events summarized by metrics such as precipitation, discharge, runoff coefficient and maximum discharge. These metrics were then used in cluster analysis to study nutrient dynamics in the Duck River, in north-western Tasmania, Australia. Minaudo et al. (2017) applied the non-linear empirical modeling method of Mather and Johnson (2014) using continuous records of turbidity and discharge to estimate high frequency phosphorus concentration values from low frequency (e.g., weekly) sampling. They then clustered storm events using sets of model coefficients that were fit to each storm event. The coefficients were re-calibrated for each cluster to obtain one set of coefficients representative of all storm events in the cluster. Mather and Johnson (2015) modeled event turbidity as a function of event discharge

using a power-law model, and performed cluster analysis on the model parameters to select the number of hysteresis loop categories, thereby avoiding *a priori* selection of the number of classes. While all of these works extract event information from two monitored variables (e.g., C and Q), none directly use the full time series (i.e., without transformation or feature extraction) associated with both variables to cluster storm events.

In this paper, we present a data-driven approach for clustering multivariate water quality time series at the event scale. We refer to this method as METS (multivariate event time series) clustering throughout the remainder of the manuscript; and show proof-of-concept using two variables: concentration (C) and discharge (Q). These time series may be visualized as trajectories in a 3-D space, namely a C-Q-T plot. Our concentration data comprise three years of high-resolution riverine suspended-sediment concentration (SSC) time series – for generalizability, referred to simply as C – collected from six watershed sites in Vermont. The efficacy of the approach is demonstrated both qualitatively, using multi-dimensional visualizations (i.e., C-Q-T plots), and quantitatively using metrics that summarize event characteristics. We also highlight the complementary nature of using METS in tandem with other analysis schemes, in this work – the C-Q hysteresis patterns of Williams (1989).

3.2 Study Area and Data

Our study area, located in the Mad River watershed (Figure 3.1) in the Lake Champlain Basin and central Green Mountains of Vermont, is the site of several ongoing geomorphic and sediment dynamics studies at the University of Vermont (Stryker et al., 2017; Wemple et al., 2017; Hamshaw et al., 2018). Continuous streamflow and suspended sediment monitoring data (SSC) were collected for more

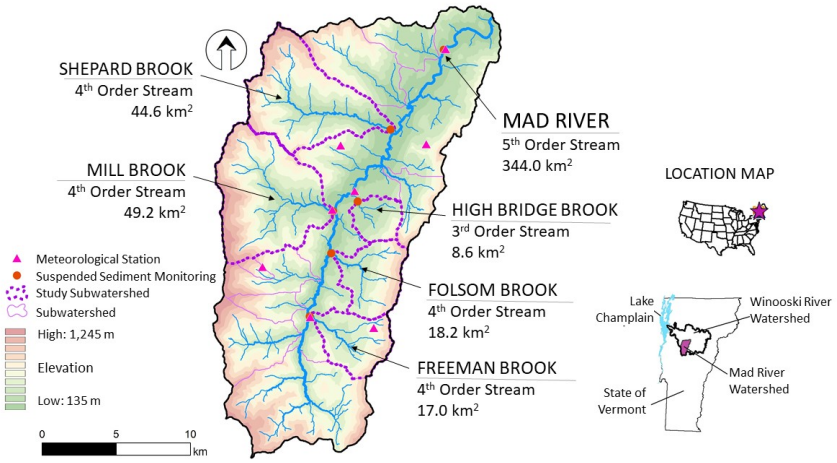


Figure 3.1: The Mad River watershed and study sub-watersheds within the Lake Champlain Basin of Vermont.

than 600 storm events in this watershed (and its five sub-watersehds) between October 19^{th}, 2012 to August 21^{th}, 2016 (Table 3.1). Hamshaw et al. (2018) used this dataset to automate and demonstrate possible refinements to the 2D (C-Q) hysteresis classifications of Williams (1989). Turbidity data were collected every 15 minutes using turbidity sensors and SSC-turbidity regression models were used to calculate SSC (see Hamshaw et al. (2018) for details). Discharge data were obtained from the United States Geological Survey (USGS) stream gauges or calculated using stage-discharge rating curves. The individual storm events were extracted from the continuous sensor records using a semi-automated approach based on thresholds to detect events and manual identification of storm end points. Meteorological data (rainfall and soil moisture) were also collected over the monitoring period and summarized into 24 storm event metrics (see Table 3.2); for full details on data collection and event delineation methodology, readers are referred to Hamshaw et al. (2018).

The Mad River watershed ranges in elevation from 132 m to 1,245 m above sea level and is predominantly forested except for the valley bottom, which features agriculture,

Table 3.1: Number of storm events and monitoring start and end dates for each watershed study site.

Site	**Number of events monitored**	**Monitoring start date**	**Monitoring end date**
Freeman Brook	54	Jun 2^{nd}, 2013	Nov 17^{th}, 2013
Folsom Brook	96	Jul 17^{th}, 2013	Sept 13^{th}, 2015
Mill Brook	158	Oct 19^{th}, 2012	Dec 23^{rd}, 2015
High Bridge Brook	41	Jun 6^{th}, 2013	Nov 17^{th}, 2013
Shepard Brook	106	Jul 18^{th}, 2013	Dec 23^{rd}, 2015
Mad River (main stem)	148	Oct 29^{th}, 2012	Aug 21^{th}, 2016
All Sites	**603**	**Oct 19^{th}, 2012**	**Aug 21^{th}, 2016**

village centers, and other developed lands (Supporting Information Table S1). The watershed has a mean annual precipitation ranging from approximately 1,100 mm along the valley floor to 1,500 mm along the upper watershed slopes (PRISM, 2019). Soils range from fine sandy loams derived from glacial till deposits in the uplands to silty loams from glacial lacustrine deposits in the lowlands. Erosional watershed processes include bank erosion, agricultural runoff, unpaved road erosion, urban storm water, and hillslope erosion. Similar to many watersheds in Vermont, reducing excessive erosion and sediment transport in the Mad River is a focus of several management efforts including stormwater management practices, streambank stabilization and river conservation.

In addition to the Mad River watershed sites, we created an expanded regional dataset by adding 190 events from three additional watersheds (Hungerford Brook, Allen Brook, and Wade Brook) in the Lake Champlain Basin to the existing ($n = 603$) Mad River events, and another 21 events from within the Mad River watershed

Table 3.2: Description of the 24 storm event metrics used in this work.

Metric	Description
Hydrograph/ Sedigraph characteristics	
T_Q	Time to peak discharge (hr)
T_{SSC}	Time to peak TSS (hr)
T_{QSSC}	Time between peak SSC and peak flow (hr)
Q_{Recess}	Difference in discharge value at the beginning and end of event
SSC_{Recess}	Difference in concentration value at the beginning and end of event
D_Q	Duration of stormflow (hr)
FI	Flood intensity
SSC_{Peak}	Peak SSC (mg/L)
HI	Hysteresis index
Antecedent conditions	
T_{LASTP}	Time since last event (hr)
A3P	3-Day antecedent precipitation (mm)
A14P	14-Day antecedent precipitation (mm)
$SM_{SHALLOW}$	Antecedent soil moisture at 10 cm depth (%)
SM_{DEEP}	Antecedent soil moisture at 50 cm depth (%)
BF_{NORM}	Drainage area normalized pre-storm baseline flow($m^3/s/km^2$)
Rainfall characteristics	
P	Total event precipitation (mm)
P_{max}	Maximum rainfall intensity (mm)
D_P	Duration of precipitation (hr)
T_{PSSC}	Time between peak SSC and rainfall center of mass (hr)
Streamflow and sediment characteristics	
BL	Basin lag
Q_{NORM}	Drainage area normalized stormflow ($m^3/s/km^2$)
Log(Q_{NORM})	Log-normal stormflow quantile (%)
SSL_{NORM}	Drainage area normalized total sediment (kg/m^2)
$FLUX_{NORM}$	Drainage area and flow normalized sediment flux ($kg/m^3/km^2$)

during the period from April $3^{rd}, 2007$ to November $25^{th}, 2016$. This results in a total of 814 storm events from nine watersheds, hereafter referred to as the "regional Vermont dataset". Hungerford Brook, Allen Brook, and Wade Brook are watersheds with ongoing monitoring efforts (Vaughan et al., 2017) that represent a spectrum of land uses (e.g., agricultural, forested, and developed, respectively) and feature varied topographic characteristics (Supporting Information Table S1). Data from these sites, and supplemental events from the Mad River do not have the corresponding hydrometeorological data metrics associated with the Mad River dataset and thus were not the focus of our primary analyses.

3.3 Methods

3.3.1 Event Time Series Processing

The sensor data collected during individual storm events are conceptualized as trajectories and may comprise *multivariate* time series of two or more variables. For example, two (univariate) time series, $TS1 = \langle V1_1, V1_2, V1_3, ..., V1_n \rangle$ and $TS2 = \langle V2_1, V2_2, V2_3 ..., V2_n \rangle$, when combined, make a bivariate time series $\mathbf{TS} = \langle (V1_1, V2_1), (V1_2, V2_2), ..., (V1_n, V2_n) \rangle$. This approach can be generalized to the multivariate case of a matrix of m variables and n time steps (Supporting Information Figure S1).

The time series in this work (discharge and SSC) were collected *in situ* using multiple environmental sensors. These data typically contain noise, have missing

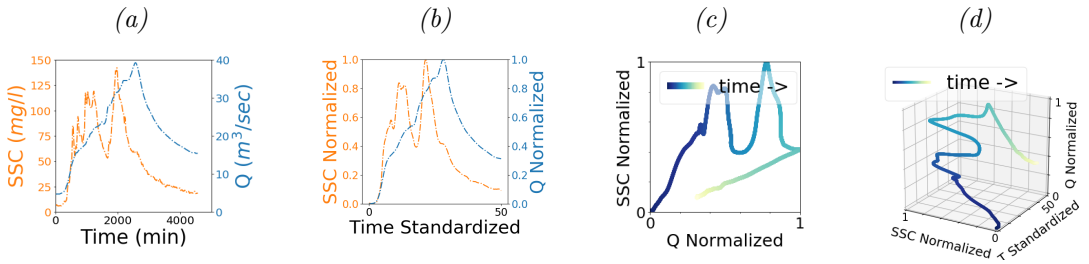


Figure 3.2: Pre-processing of (a) raw C and Q time series, (b) smoothed and normalized C and Q time series, and the resulting (c) C-Q plot, and (d) C-Q-T plot for an individual (delineated) storm event.

values, and often require pre-processing (i.e., filtering) to extract general trends in the C-Q relationship. In addition, because of our interest in comparing C-Q relationships across hydrological events, we normalized both the length of the time series as well as the magnitude of each variable individually over each event (Figure 3.2), as is commonly done in C-Q analyses. Pre-processing steps were performed as follows:

Smoothing: To reduce noise, the discharge and concentration time series were smoothed using the Savitsky-Golay Filter (Savitzky and Golay, 1964). We selected a third-order, 21-step filter for the Mad River (main stem) and a fourth-order, 13-step filter for each of the five sub-watersheds. To preserve the peaks and overall shape of the event data, the filter order and step size were selected based on visual inspection of the resulting event time series in a manner similar to Hamshaw et al. (2018).

Standardization of event length: Discharge and concentration time series were re-scaled to a uniform length of 50 time steps for all events using univariate spline fitting (Dierckx, 1993). The number 50 was selected empirically as the minimum number of data points that preserves the shape and characteristics of

the event time series. Standardizing all events to have the same length ensured that clustering was not affected by the duration of the event but by the relative rate of change of C-Q variables. We note that this re-sampling was performed separately from the calculation involving event metrics (Table 3.2) based on the original data.

Normalization of magnitude: The discharge and concentration time series were scaled individually to values between 0 and 1. This ensured that the clustering is not affected by the magnitude of the individual time series but by the orientation of change (e.g., clockwise and counter-clockwise), and the shape (e.g., linear, convex and concave). Normalizing the magnitude of variables is common for a meaningful comparison between time series (Rakthanmanon et al., 2012).

3.3.2 Concentration-discharge (C-Q) Hysteresis Classification

Each hydrological event in our dataset was categorized visually (by two or more domain experts) into one of the six hysteresis classes (Figure 3.3) of Williams (1989). Class I represents linear C-Q relationships that show little hysteretic behavior, whereas Class II and Class III represent clockwise and counter-clockwise hysteretic behaviors, respectively. A C-Q plot exhibiting a linear relationship followed by a clockwise loop is indicative of Class IV behavior. These patterns could reasonably be considered a special case of Class II (clockwise hysteresis); and rarely are studied

as a separate hysteresis category (Malutta et al., 2020). The figure-eight loops are represented as Class V. Events that do not fall into any of these five classes are placed into a class labeled "Complex".

3.3.3 Multivariate Time Series Clustering

Clustering of the multivariate time series data at the storm event scale was a first step in exploring linkages between storm event responses (i.e., C-Q dynamics) and watershed processes. To this end, a number of clustering methods were investigated. Paparrizos and Gravano (2017) conducted extensive benchmark tests using four clustering algorithms (partitional, hierarchical, spectral, and density-based) and three distance measures – Euclidean distance, dynamic time warping of Sakoe and

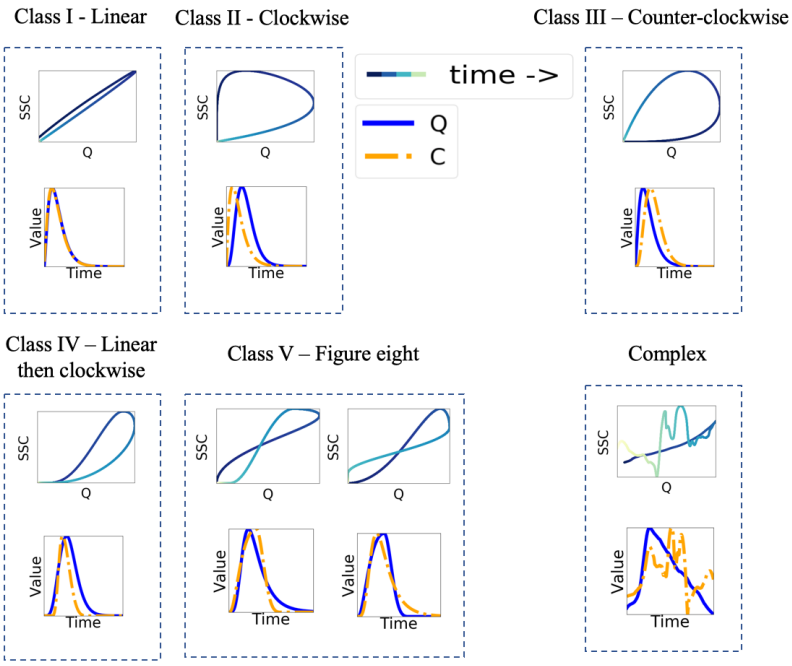


Figure 3.3: Six class scheme for concentration-discharge hysteresis loops (top panels) and corresponding hydrographs and sedigraphs (lower panels, solid and dot-dashed lines, respectively).

Chiba (1978), and shape-based (Paparrizos and Gravano, 2016). All of the datasets (85 in total) available in the University of California at Riverside (UCR) time series archive (Dau et al., 2018) at the time of their publication were used in the benchmark; they identified K-medoids with dynamic time warping (DTW) (discussed in Section 3.3.3 and Section 3.3.3, respectively) as having achieved the highest adjusted Rand index across the greatest number of datasets. Leveraging their work, we conducted additional benchmark tests using the four algorithms on their short list — TADPole (Begum et al., 2015), K-shape (Paparrizos and Gravano, 2016), K-medoids with DTW, and K-medoids with Euclidean. Using all datasets (currently 128 in total) available in the UCR time series archive (Dau et al., 2018), we also found that K-medoids with DTW achieved the highest adjusted Rand index across the greatest number of datasets. All of the event time series data in UCR archive were pre-processed as outlined in Section 3.3.1 to avoid unexpected consequences that might result from treating benchmark data differently from our hydrological event dataset.

K-medoids Clustering Algorithm

K-medoids is a variant of the popular K-means (Wu et al., 2007), in which the cluster centroids are observation points (called "medoids") as opposed to coordinates as in K-means. These medoids are mapped from a multivariate time series of length n (i.e., $t_1, t_2, ..., t_n$) to vectors of the multiple variables (i.e., $V1, V2, ..., Vm$) at each time step t_i. Like K-means, the K-medoids algorithm is iterative (Supporting Information Algorithm S1) where the initial K medoids are selected randomly. The algorithm

has two phases: Phase 1 assigns observation points to clusters (Line 3); and Phase 2 calculates new medoids for each cluster (Line 4). In Phase 1, the distance between all observation points and each of the medoids is calculated, and each observation point is assigned to the closest medoid. In Phase 2, a new medoid is selected from each cluster by finding the observation point that minimizes the sum of squared distances (i.e., sum of squared errors) to all other observation points in that cluster. These two phases are repeated for a given number of iterations or until there is no change in the medoid selection. Algorithm S1 in Supporting Information was implemented in Python (version 3.6.1); the source codes may be found at GitHub (Javed, 2019b).

For a given dataset, the optimal number of clusters may vary depending on the research question/objective. In this study, the elbow method guided the selection of the "optimal" number of clusters. This method consists of plotting the sum of squared errors (SSEs) against an increasing number of K clusters. An optimal value for K is selected (visually) as the value for which further increases in K result in diminishing reduction in SSE, thus creating the onset of the plateau.

Dynamic Time Warping

The K-medoids clustering algorithm used a variant of dynamic time warping (DTW) to calculate the distance between two multivariate times series. Originally introduced for speech recognition (Sakoe and Chiba, 1978), DTW is arguably the most popular distance measure for time series clustering, and is particularly appealing for sensor data generated during hydrological events because of (i) the challenges associated with defining the beginning and end of an event (i.e., the ambiguity inherent in event

delineation), and (ii) the noise present in the sensor data (e.g., variability in readings due to sensor interference from debris, maintenance activities, and temporary fouling.)

Figures 3.4a and 3.4b illustrate how distance between two time series ($T1$ in red and $T2$ in blue) is calculated using the more common Euclidean distance compared with DTW. While Euclidean distance uses a one-to-one alignment, DTW employs a one-to-many alignment that enables a warping of the time dimension to minimize the distance between the two time series. As such, DTW can optimize alignment, both global alignment (by shifting the entire time series left or right) and local alignment (by stretching or squeezing parts of time series). Paparrizos and Gravano (2016) showed that the best accuracy (as measured by the Rand index) is obtained when DTW is constrained to a limited window size. Multiple window size constraints ranging from 0% to 100% were tested to cluster our Mad River dataset. Based on a preliminary qualitative analysis of event visualizations, a window size constraint of 10% was selected for our analysis. Constraining the window size to 10% of the observation data is usually considered adequate for real applications (Ratanamahatana and Keogh, 2004); and it accommodates minor differences in timing between similar hydrological events, as is often the case when delineating the end of an event proves challenging.

Aligning two time series, $T1$ of length a and $T2$ of length b, using DTW involves creating an $a \times b$ matrix, D, where the element $D[i, j]$ is the square of the Euclidean distance, $d(t1_i, t2_j)^2$, $d(\cdot, \cdot)$ is the Euclidean distance, $t1_i$ is the ith point of $T1$, and $t2_j$ is the jth point of $T2$. A warping path P is defined as the sequence of matrix elements that are mapped between $T1$ and $T2$ (see Figures 3.4c and 3.4d). This

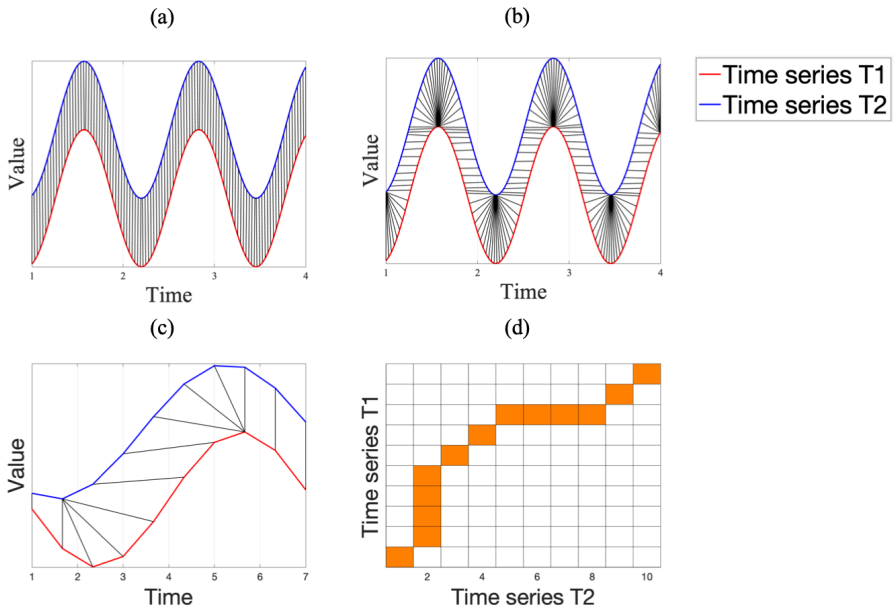


Figure 3.4: The top row illustrates the alignment between two times series for calculating distance in (a) Euclidean (one-to-one) and (b) dynamic time warping (one-to-many); Bottom row illustrates an optimal (c) alignment of each point in time series T1 and time series T2 (shown with black lines) and (d) warping path, i.e., optimal alignment of time series $T1$ (red) and $T2$ (blue), where each matrix cell (i, j) is the distance between ith element of $T1$ and jth element of $T2$; the DTW distance is the sum of the distances along the optimal path shown in orange.

warping path must satisfy the following three conditions:

1. Every point from $T1$ must be aligned with one or more points from $T2$, and vice versa.

2. The first and last points of $T1$ and $T2$ must align, meaning the warping path must start and finish at diagonally opposite corner cells of the optimal warping matrix.

3. No cross-alignment is allowed, that is, the path must increase monotonically within the matrix.

For all paths that satisfy the three conditions above, DTW finds a path that

minimizes the distance calculated as in Equation 3.1 (Shokoohi-Yekta and Keogh, 2015):

$$\mathrm{DTW}(T1, T2) = \min_{P} \sqrt{\sum_{(i,j)\in P} D[i,j]}, \tag{3.1}$$

Algorithm S2 in Supporting Information outlines the procedure for calculating this minimum distance using dynamic programming method (Bellman, 1957).

The environmental sensor data in this proof-of-concept are bivariate, representing water quality concentration and stream discharge time series. There are two DTW variants – DTW-independent (DTW-I) and DTW-dependent (DTW-D). In DTW-I, the distance between $T1$ and $T2$ is the sum of distances calculated separately for each variable (by invoking the DTW algorithm for each variable). Whereas in DTW-D, $T1$ and $T2$ are handled as *multivariate* time series; and the DTW algorithm is invoked only once. Because of the strong dependency between discharge and concentration in this work, DTW-D is used. The source code, implemented in Python (version 3.6.1), may be found at GitHub (Javed, 2019a).

3.3.4 Generating Synthetic Hydrograph and Concentration-graph Data

Synthetic multivariate times series "event data" were generated using eight conceptual hydrographs and two conceptual concentration graphs (Figure 3.5), and then combined to produce a set of heterogenous, albeit simplified, hydrographs and sedigraphs (concentration graphs). A stochastic generator was designed to produce

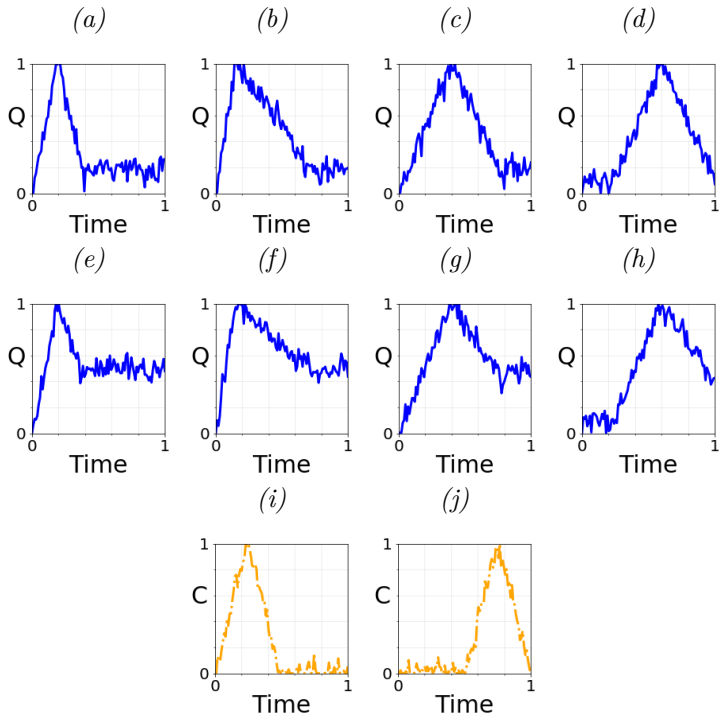


Figure 3.5: Example synthetic hydrographs and concentration graphs generated from eight conceptual hydrograph types: (a) flashy, early peak – return to baseline flow, (b) early peak – slow return to baseline flow, (c) mid-peak – return to baseline flow, (d) delayed rise to peak – return to baseline flow, (e) flashy, early peak – incomplete return to baseline flow, (f) early peak – slower incomplete return to baseline flow, (g) mid-peak – incomplete return to baseline flow, and (h) delayed rise to peak – incomplete return to baseline flow, and two conceptual concentration graphs: (i) early peak and (j) late peak.

synthetic data with sensor noise. Random samples were drawn from a normal (Gaussian) distribution with a mean of 0.00 and standard deviation of 0.05 and added to the discharge and concentration values at each time step in order to simulate noise. When combining each of the eight synthetic hydrograps with the two concentration-graphs, sixteen synthetic storm event types can be produced. These combined event types can be labeled and used as "ground truth" events to help assess and validate the methodology.

Five control parameters, ranging from 0 to 1, were used to generate the synthetic graphs: time-to-peak, duration-of-peak, delay, recess, and initial baseline conditions.

Time-to-peak controls the timing for the concentration/discharge values to reach the peak (normalized value of 1); duration-of-peak controls the duration of flow above baseline conditions; delay controls the time at which the value (either discharge or concentration) begins to rise in magnitude above the baseline conditions; recess controls the degree to which event concentration/discharge values return to the baseline conditions; and initial baseline controls the minimum value of the flow over an event. Parameter values for generating each type of synthetic graph (hydrograph and concentration-graph) were determined qualitatively based on re-production of simplified yet realistic approximation of typical hydrographs and sedigraphs observed in our study watershed (Supporting Information Table S2).

3.3.5 Measures for Assessing Clustering Performance

We used the *Hopkins Statistic* to measure the clustering tendency of our three datasets (i.e., the synthetic dataset, the Mad River dataset and the expanded regional Vermont dataset). The statistic value ranges from 0 to 1, where 1 indicates a high tendency to cluster and 0 indicates uniformly distributed data (Banerjee and Dave, 2004). Additionally, transformed variables (those representing the 24 storm event metrics of Table 3.2) were examined post-clustering to see whether these event metrics had 1) any association with clusters or 2) statistical power to differentiate between clusters using One-way Analysis of Variance (ANOVA) followed by Tukey Honest Significant Differences (HSD) tests between individual group means. For those variables (or

their transformations) that were not normally distributed, nonparametric methods were applied (Kruskal-Wallis). Lastly, *Z-score* values were calculated for each of the 24 storm event metrics of Table 3.2 to identify feature importance associated with cluster differences. The Z-score represents the distance of an individual storm metric from the population mean (measured in terms of standard-deviation).

3.4 Results

3.4.1 Using Synthetic Data to Validate Methodologies

To help validate the METS clustering approach, we generated 800 synthetic storm events, equally distributed among the sixteen possible combinations (see Section 3.3.4). As one might expect, the synthetic data had a high clustering tendency (Hopkins statistic of 1.00); and the optimal number of clusters, determined using elbow method as $K = 16$ (see Figure 3.6a), matched the intended synthetic design (16 event types). Examples of synthetic events from each of the 16 event classes are shown in Figure 3.7. Despite the presence of stochastically generated noise, the synthetic dataset clustered with 100% accuracy using K-medoids with DTW (i.e., clusters were identical to the ground truth).

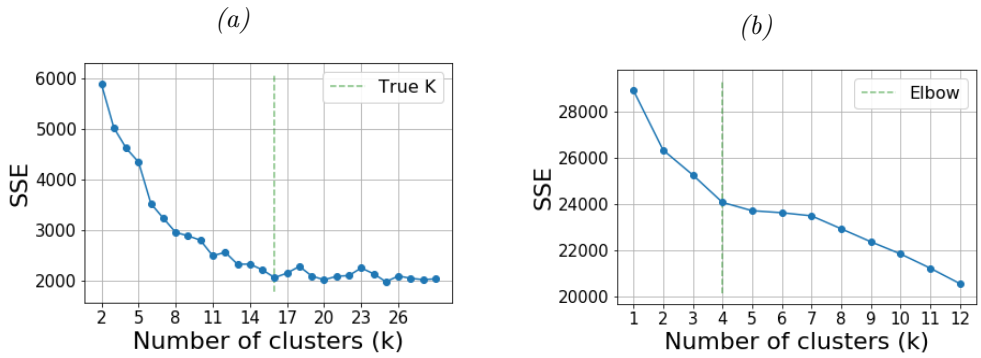


Figure 3.6: Sum of squared errors (SSE) for different number of clusters from (a) the synthetic storm event dataset (elbow point at K=16) and (b) the Mad River storm event dataset (elbow point at K=4).

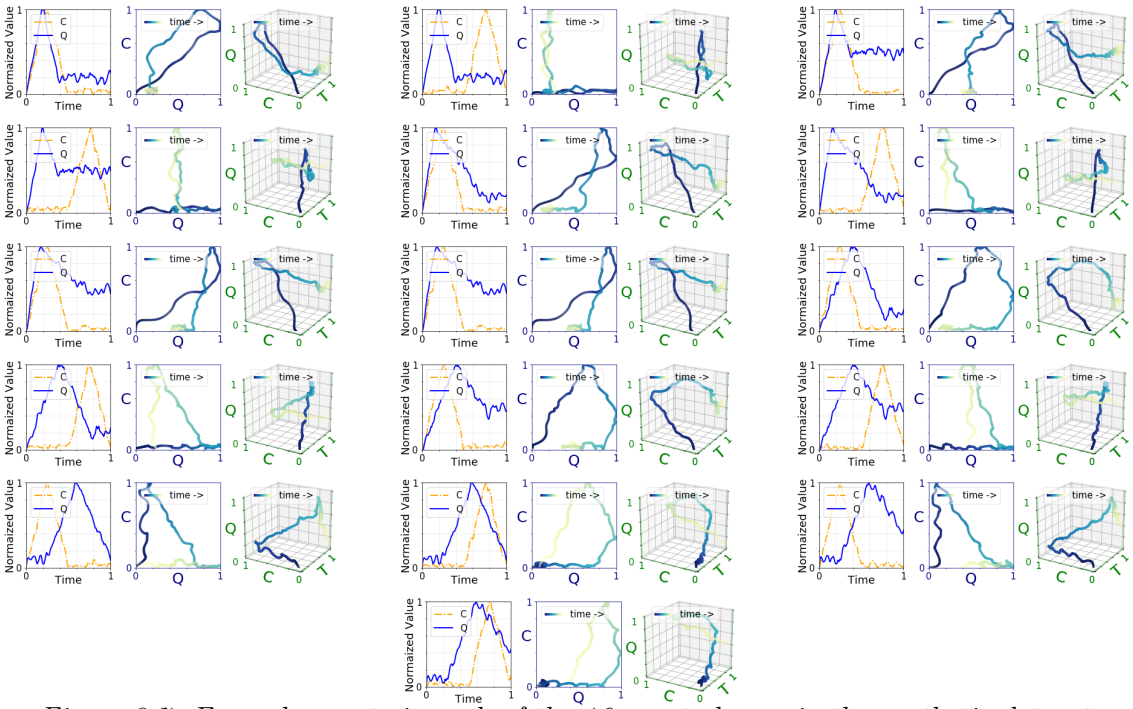
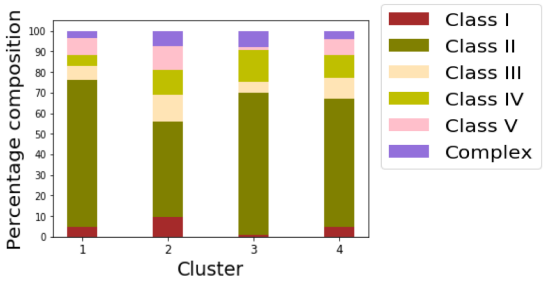


Figure 3.7: Example events in each of the 16 event classes in the synthetic dataset.

3.4.2 Application of METS to the Mad River Dataset

In applying the METS clustering to the 603 Mad River storm events, we identified $K = 4$ event clusters with distinct SSC and Q responses (see the plateau in the elbow



Cluster	Class I	Class II	Class III	Class IV	Class V	Complex	Total
1	11	167	16	12	20	8	234
2	12	58	16	15	15	9	125
3	1	80	6	18	2	9	116
4	6	80	13	14	10	5	128
Total	30	385	51	59	47	31	603

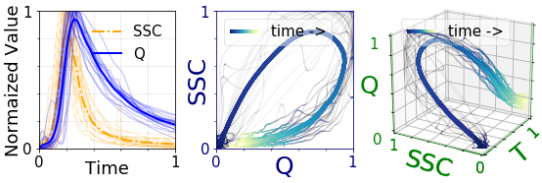
Figure 3.8: Distribution of hysteresis loop classes over METS clusters.

plot of Figure 3.6b). Approximately one third of the events (n = 234) fell into cluster 1, with each of the three remaining clusters having between 116 and 128 events (see Figure 3.8). Unlike the synthetic dataset, the optimal number of clusters for the Mad River dataset, any real dataset for that matter, will never be known with any degree of certainty. However, these data have a Hopkins test statistic of 0.96 indicating they are highly clusterable. We first explored whether a relationship existed between the four METS clusters and the six-class hysteresis scheme presented in Section 3.3.2. We found little association between the two as the confusion matrix and cluster distribution of Figure 3.8 show the six classes to be fairly evenly distributed across the four METS clusters.

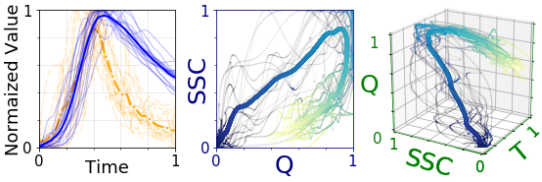
Qualitative interpretation of METS clusters using event visualizations

Finding little relationship between the METS clustering and the hysteresis classification, we further investigated the characteristics associated with combined

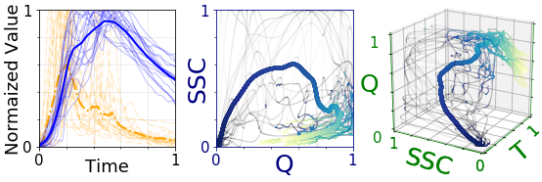
(a) Cluster 1. n = 234



(b) Cluster 2. n = 125



(c) Cluster 3. n = 116



(d) Cluster 4. n = 128

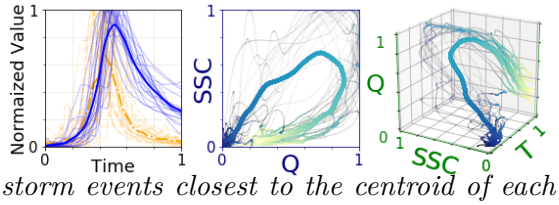


Figure 3.9: Mad River storm events closest to the centroid of each of the $K = 4$ clusters, superimposed on a single graph with the mean value plotted as a solid line — (a) cluster 1 events have a broad clockwise hysteresis pattern featuring an early and relatively brief duration of high SSC, (b) cluster 2 events have a narrow clockwise hysteresis loop and broad sedigraphs and hydrographs with streamflows that do not fully return to baseline levels, (c) cluster 3 events have flashier and sometimes multi-peaked sedigraphs that are shorter in duration, and (d) cluster 4 have a delayed rise of hydrograph and sedigraph, and typically more aligned.

hydrograph and sedigraph trajectories of the METS clusters using multiple visualization approaches. To visualize overall trends, we superimposed 20 storm events closest to the centroid of each of the four METS clusters onto single plots (Figure 3.9); mean values are plotted as solid lines. Additionally, examples of the event times series, C-Q hysteresis plots, and 3-dimensional C-Q-T plots for each cluster are provided in Figure 3.10. In general, the METS cluster 1 events

(Figure 3.9a and Figure 3.10a) have broad clockwise hysteresis patterns with an early, and relatively brief duration of high SSC. The hydrographs are flashy, rise quickly and return nearly to baseline flows. Cluster 2 events typically have a more narrow hysteresis loop compared to cluster 1 and broad (less flashy) sedigraphs and hydrographs with streamflows that do not fully return to the baseline levels (Figure 3.9b and Figure 3.10b). Cluster 3 events are similar to cluster 2, but exhibit flashier and sometimes multi-peaked sedigraphs that are shorter in duration (Figure 3.9c and Figure 3.10c). Multi-peaked events sometimes exhibit compound behavior including, for example, portions of clockwise hysteresis loops and no hysteretic behavior (linear relationships). Cluster 4 events typically have a delay in the rise of the hydrograph and sedigraph, and typically more aligned (Figure 3.9d and Figure 3.10d). In contrast to cluster 2 and 3 events, the hydrographs of cluster 4 also tend to return to near baseline levels.

Statistical Analysis of METS clusters

Of the 24 storm event metrics in Table 3.2, 19 metrics had significantly different mean values for at least one of the METS clusters. The reader should bear in mind that these event metrics were not used as input to either the METS clustering algorithm or the hysteresis classification scheme. Both the METS clusters and hysteresis classes have event metrics with good discriminatory power; but there was little overlap for a given metric. For instance, two of the metrics shaded in Table 3.3 (e.g., SSC_{Peak} and the difference in discharge values at the beginning and end of an event (Q_{Recess})) show an ability to discriminate between the clusters generated by METS, but little

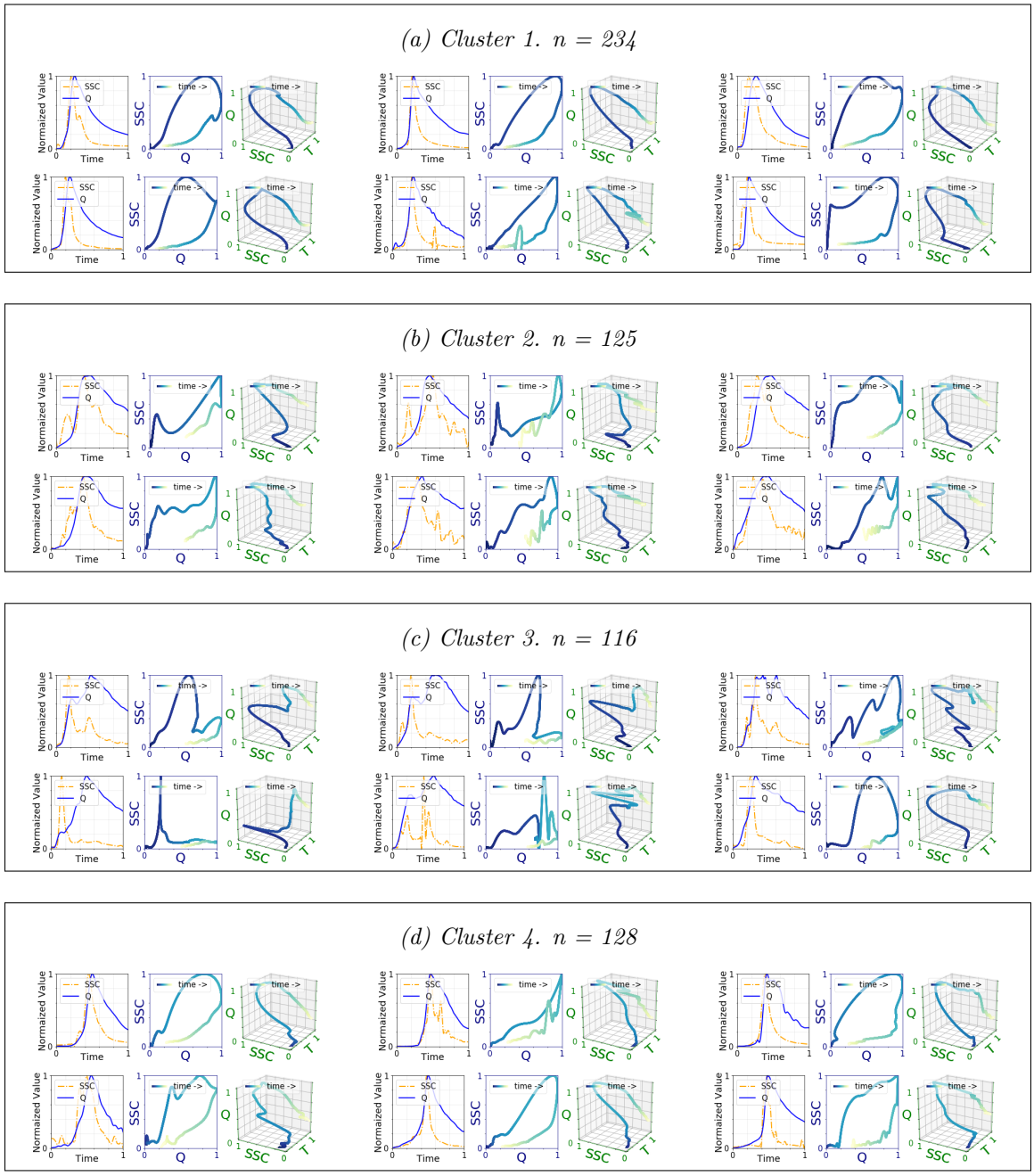


Figure 3.10: Six storm events closest to the centroid of the four Mad River dataset METS clusters ($K = 4$, $N = 603$) — (a) cluster 1 events have a broad clockwise hysteresis pattern featuring an early and relatively brief duration of high SSC, (b) cluster 2 events have a narrow clockwise hysteresis loop and broad sedigraphs and hydrographs with streamflows that do not fully return to baseline levels, (c) cluster 3 events have flashier and sometimes multi-peaked sedigraphs that are shorter in duration, and (d) cluster 4 have a delayed rise of hydrograph and sedigraph, and typically more aligned.

Table 3.3: Result of post-hoc Tukey HSD test ($\alpha = 0.05$) for all pairwise comparisons of hydrograph/sedigraph related storm event metrics. Within each metric, if two classes/clusters do not share the same letter, the metric means are significantly different.Shaded columns are highlighted to show examples of metrics distinguished well by METS, but not by hysteresis classes (light shading) and metrics discriminated well by hysteresis classes (dark shading).

Hydrograph/Sedigraph Characteristics									
Metric	T_Q	T_{SSC}	T_{QSSC}	Q_{Recess}	SSC_{Recess}	D_Q	FI	SSC_{Peak}	HI
METS clusters									
cluster 1	a	a	a	a	a	a	a	a	a
cluster 2	b	b	a	b	b	a b	b	b	b
cluster 3	b	c	b	c	a	a	b	b c	b
cluster 4	c	b	a	d	c	b	b	a c	b
Hysteresis classes									
Class I	a b	a b	a	a b	a b	a b	a b	a	a
Class II	a	a	b	a	a	a	a b	a	b
Class III	a	a	c	a	b	a b	a b	a	c
Class IV	a b	a b	a b	b	a	a b	a	a	d
Class V	a	a	a	a	a	a b	b	a	a
Complex	b	b	a b	a b	a	b	a b	a	a

statistical power to discriminate between the six classes of the hysteresis classification method. In contrast, both the hysteresis index (HI) and time between peak SSC and peak flow (T_{QSSC}) show power to discriminate between the hysteresis classes, but not the MET clusters (Table 3.3). Similar differences in discriminatory power were observed in metrics related to antecedent conditions, rainfall characteristics, and streamflow/sediment characteristics (Supporting Information Table S3 to Table S5).

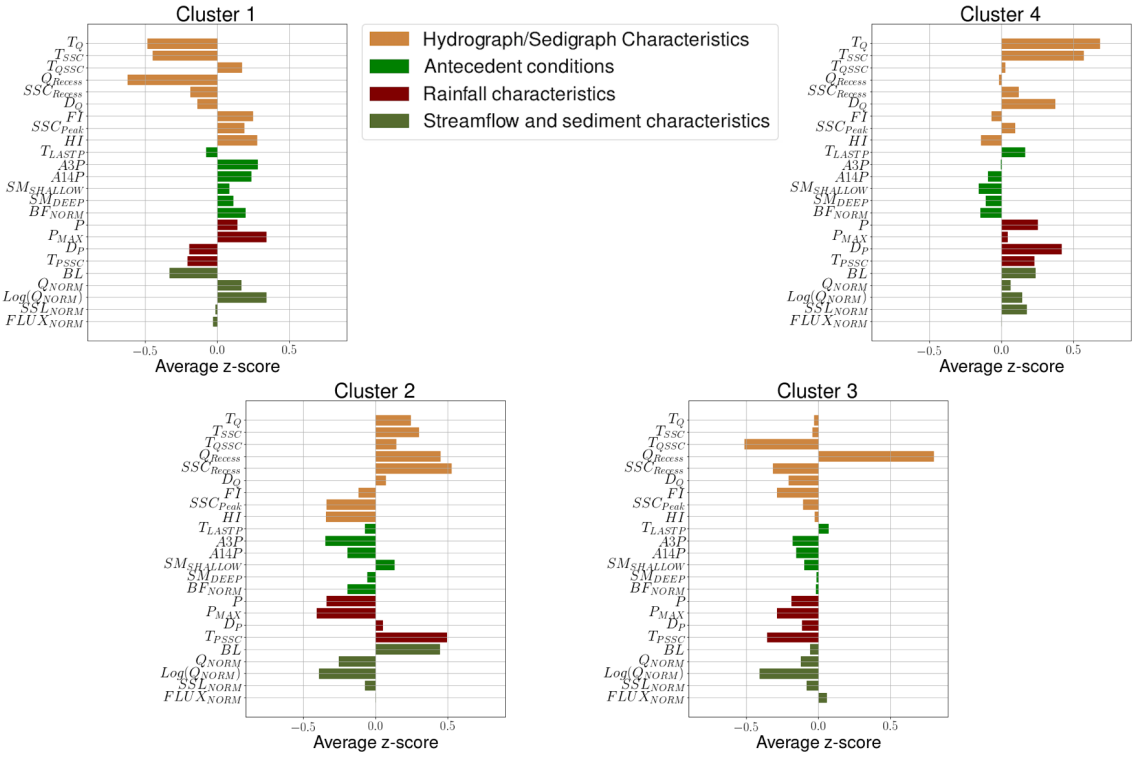


Figure 3.11: Typical hydrometeorological characteristics of METS clusters as represented by storm event Z-score metrics for each of the four clusters.

Next, we explored the hydrometeorotological factors associated with the four METS clusters using event metric Z-score values. Again, these event metrics were not used as input to the clustering algorithm, but as a means to study linkages between these characteristics and the resulting clusters. The storm events of cluster 1 have greater amounts of precipitation (positive Z-score for P and P_{Max}) and wetter antecedent conditions exhibited by higher mean BF_{Norm}, SM_{Deep},$SM_{Shallow}$, $A3P$ and $A14P$. In general, these factors are associated with higher stream discharge as confirmed by the positive Z-score for Log (Q_{Norm}), Q_{Norm}, and FI (flood intensity) as well as higher peak SSC values. Other notable characteristics include hydrographs that return to baseline flow (negative Z-score for Q_{Recess}), and a rapid rise in the

sedigraph and hydrograph (negative Z-score for T_{SSC} and T_Q) and positive Z-score for HI, which translate to a 2D hysteresis that is dominated by a broad clockwise pattern (observed in Figure 3.9a and Figure 3.10a).

Cluster 2 is associated with smaller precipitation events (negative Z-score for P and P_{Max}) and drier antecedent conditions (negative BF_{Norm}, SM_{Deep}, $A3P$ and $A14P$ Z-scores), both resulting in lower stream discharge (negative Log (Q_{Norm}), Q_{Norm}, and FI Z-scores). These events also have positive Q_{Recess} and SSC_{Recess} Z-score values. These two metrics were designed to capture whether streamflow and SSC return to baseline levels; positive scores are associated with events that do not return to base levels (Figure 3.9b and Figure 3.10b). Additional characteristics include lower peak SSC concentrations and negative Z-scores for BL (indicative of watersheds that respond more slowly to a rainfall event), and a longer duration between the peak SSC and center of mass for rainfall (positive Z-score for T_{PSSC}). The latter translates to hysteresis patterns with more narrow loop, which is confirmed visually (Figure 3.9b and Figure 3.10b), and by the negative Z-score for hysteresis index.

Cluster 3 events have a rapid rise in both streamflow and SSC (Figure 3.9c and Figure 3.10c) and are associated with a positive Z-scores for Q_{Recess} and negative for SSC_{Recess}, which is indicative of sedigraphs that return to base levels and hydrographs that do not. The sedigraph is also often characterized by multiple peaks; and in general, there is a short duration between the peak SSC and the center of mass for rainfall (negative Z-score for T_{PSSC}) as well as between the peak SSC and peak discharge (negative T_{QSSC}). In addition, these events have lower precipitation (negative Z-scores for P and P_{Max}) and stream discharge (negative Log (Q_{Norm}),

Q_{Norm}, and FI), as well as Z-scores that approach zero for BF_{Norm}, SM_{Deep}, $SM_{Shallow}$, $A3P$ and $A14P$, which indicate average antecedent conditions.

Lastly, cluster 4 events are associated with higher precipitation (positive Z-score for P) that are longer in duration (positive Z-score for D_P); however, these events have less intense rainfall (near zero Z-score for P_{Max}), and are associated with average to fairly dry antecedent conditions (i.e., slightly negative Z-score values for BF_{Norm}, SM_{Deep}, $SM_{Shallow}$, $A3P$ and $A14P$), all of which results in near average streamflows (near zero Z-score for Log (Q_{Norm}), Q_{Norm}, and FI). Other event characteristics include a long time to peak SSC and Q (positive Z-score for T_{SSC} and T_Q) and larger amounts of sediment transport during events (positive SSL_{Norm}).

3.4.3 Effects of Additional Watersheds on METS Clustering

The number and type of event clusters/classes are dependent on geographic range of study. In re-running the METS analysis on the expanded regional Vermont dataset, the number of clusters increased from $K = 4$ to $K = 9$ (Supporting Information Figure S2). This is not surprising given the differences, particularly in topography and land use, associated with the added watersheds. Hungerford Brook, for instance, is a low gradient agricultural basin, while Allen Brook drains a highly developed suburban area (Supporting Information Table S1). The METS results show the expanded dataset cluster 5 to have a substantially large number (54%) of counter-clockwise hysteresis loops, which correspond to events where the sedigraph peaks after the

hydrograph (hysteresis Class III), and no events that are clockwise (hysteresis Class II or Class IV) (Supporting Information Figure 3.12 and Table S6).

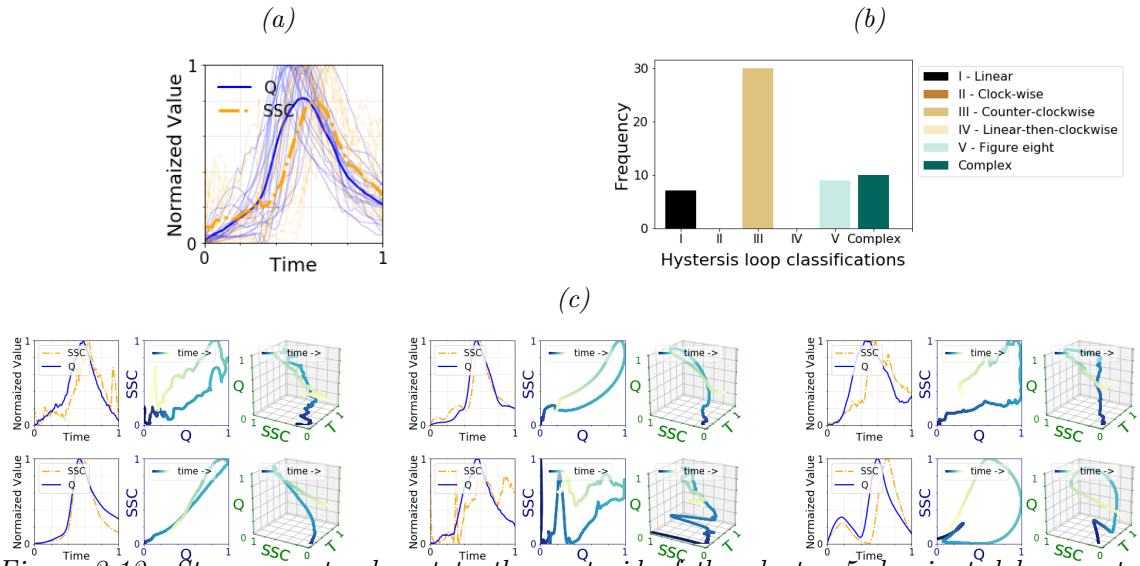


Figure 3.12: Storm events closest to the centroid of the cluster 5 dominated by counter clockwise hysteresis type events (when $K = 9$) in the expanded regional Vermont dataset, discovered by including more watersheds: (a) all 56 events in cluster 5 superimposed, with the mean plotted as a solid line, (b) distribution of cluster by hysteresis loop classification, and (c) six events closest to the centroid of the cluster ($n = 56$).

3.5 Discussion

We present a new clustering approach within the broader discipline of event-based studies — one that leverages the temporal information in two or more time series for the purpose of grouping or identifying similar events — in this manuscript, a *hydrological* event comprising hydrograph and sedigraph data modeled as three-dimensional C-Q-T trajectories. This contrasts with current hydrological event approaches that either collapse the time dimension (e.g., 2D hysteresis pattern analysis of Lloyd et al. (2016b)) or focus on the response of a single variable such as the DTW clustering approach of Dupas et al. (2015); the latter re-scales events using

a single (ideal) hydrograph and then clusters the concentration response. While these approaches are important to a variety of research applications, these 2-D hysteresis methodologies lose the temporal information, while the latter requires a rescaling of the C-Q variables. The multivariate version of DTW-D used in the METS clustering of this manuscript is designed to extract relationships between the time series of two or more variables, resulting in a dataset partitioning that is dissimilar and complementary to existing hysteresis methods.

3.5.1 Effects of Regional Scale on METS Clustering.

Our motivations for limiting the primary analysis to the Mad River watershed were two-fold. First, meteorological data were not available for the additional watersheds; and secondly, we wanted, at least initially, to control for certain watershed characteristics such as topography and land use (e.g., the Mad River has primarily two land use types - forest and agriculture). In this single watershed study, we identified four predominant clusters for hydrological events occurring between the period from 2013 and 2016, with one cluster type occurring most frequently (38%), and 64% of the events categorized as clockwise patterns. This relatively small number of event types (i.e., four clusters) might be expected, given the uniformity of watershed characteristics across the six Mad River monitoring sites; as this is similar in number to other event analyses from single study areas. Bende-Michl et al. (2013) identified 3-4 cluster in a study on nutrient dynamics; Mather and Johnson (2015) identified 5-7

clusters when analyzing C-Q loops; and 3 nutrient-event response types were identified in the work of Dupas et al. (2015). In general, there is a great deal of interest and merit in tracking the change in both the number and type of event responses within a single study area, particularly for example, when monitoring in-stream changes prior to and after restoration efforts. However, other monitoring applications may require tracking changes across watersheds at larger geographical scale; and one might expect the number of clusters (event types) to increase with the geographic range of study as demonstrated in Section 3.4.3.

Regardless of regional scale, we found the METS clustering to be heavily influenced by the degree to which both of the time series (SSC and Q) return (or not) to base levels at the end of the event. This was evidenced both visually (Figure 3.10) and by the significance of the SSC_{Recess} and Q_{Recess} metrics (Table 3.3 and Figure 3.11). From a hydrological perspective, the rate and degree of recession (return to baseline flow and background concentration levels) are important indicators of soil moisture, groundwater elevations, and the resulting hydrological flowpaths. Classification schemes based on the shape and direction of hysteresis do not necessarily capture this "return to baseline conditions" behavior because the overall C-Q patterns are primarily driven by the middle portion of the hydrograph-sedigraph (i.e. largest offset between C-Q) rather than differences between the times series at the start or end of the event. The ability of the METS clustering to capture this return-to-baseline conditions phenomena, in addition to other metrics, holds promise for many applications (e.g., model validation) used in forecasting floods, water quality monitoring, watershed similarity studies, and detecting change in

watershed functions.

3.5.2 Leveraging Methodological Strengths to Group Events

The post-cluster analysis performed on event metrics (hydrological and meteorological metrics in Table 3.2) was an attempt to explore which factors (i.e., characteristics associated with the event time series) might be driving the METS clustering, bearing in mind that these metrics were not used as inputs to the clustering analysis itself. Prior event-based hydro-meteorological studies have successfully used this type of post-statistical analysis to tease out factors important in discriminating between (or correlated with) event groupings. Examples include the classifying of event hysteresis patterns to study erosional processes (Seeger et al., 2004; Nadal-Romero et al., 2008; Sherriff et al., 2016; Hamshaw et al., 2018).

Here, we highlight some key results from our post-cluster statistical analysis, particularly the event metric with statistically significant differences across the METS clustering and/or hysteresis classification. First, while the event hysteresis index (HI) was identified, not surprisingly, as important for differentiating between the hysteresis class types (see Table 3.3 in Supporting Information), the temporal hydrograph and sedigraph metrics (e.g., time to peaks – T_Q, and T_{SSC}), as well as the degree to which both time series return to baseline conditions (Q_{Recess} and SSC_{Recess}) were not identified as important drivers. In contrast, these four metrics as well as the Peak SSC (SSC_{Peak}), duration of stormflow (D_Q) and antecedent precipitation

metrics (Section 3.4.2) were identified as important for differentiating between the METS-based clusters (Table 3.3 and Supporting Information Table S3).

3.5.3 Using Methods in Tandem to Leverage Strengths

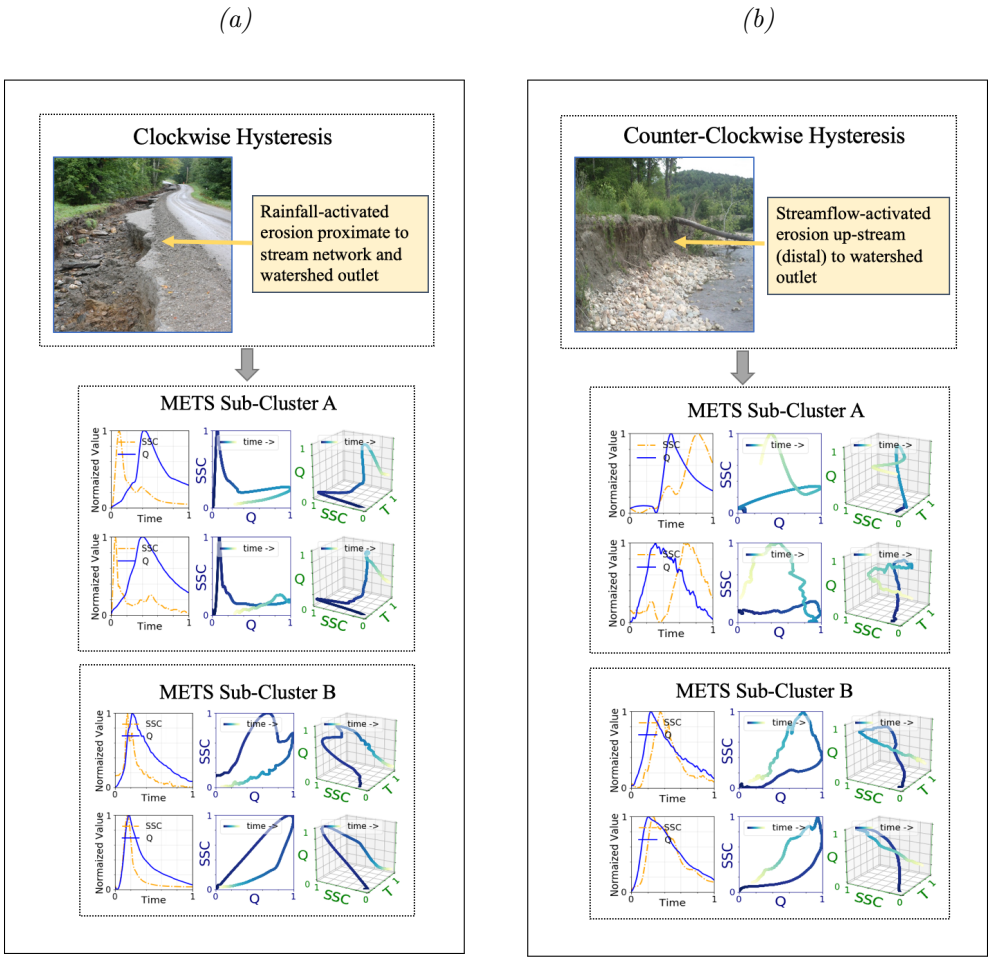


Figure 3.13: Application of METS after pre-classifying events based on hysteresis directions of (a) clockwise hysteresis and (b) counter clockwise hysteresis that can correspond to general proximity and timing of erosion source activation. METS clustering further partitions these hysteresis classes into sub-clusters (visualized as two example events) distinguishable by different hydrograph and sedigraph characteristics. Photos from observed, active erosion sources within the Mad River watershed.

Each of the clustering and classification approaches have unique strengths and weaknesses; and the post-statistical analyses (e.g., Tukey HSD test and Z-scores of Section 3.4.2) provide some guidance on method selection that best aligns with manager or stakeholder goals. However, using more than one method in tandem may help to leverage methodological strengths. For example, in event-based suspended sediment studies — those aimed at identifying the proximity of riverine erosion sources, a two-phased approach may add value. Let's consider our expanded dataset in which more than two thirds of the events have clockwise hysteresis patterns. A first phase might use hysteresis classification to prioritize the clockwise versus counter-clockwise nature of the hysteresis patterns, as the direction embeds key process information. This Phase I classification could then be further partitioned into subgroups (via METS methodology) to help refine the understanding of watershed processes.

To highlight the potential of such an approach, we applied the 2-D hysteresis analysis and METS clustering in tandem using the expanded dataset of Section 3.4.3. In Phase I, hydrological events were classified (e.g., into clockwise and counter-clockwise groups) based on their hysteresis patterns; and in Phase II, the METS clustering was applied to each of the Phase I classes, respectively (Figure 3.13 and Supporting Information Figure S3 and Figure S4). Clockwise hysteresis patterns are typically indicative of erosion sources (e.g., gullies or rills) that are located very close to the monitoring site. Whereas the events in the counter-clockwise group are characterized by hydrographs that occur (and peak) prior to the accompanying sedigraphs. These are often indicative of more distal sediment sources (e.g., upstream

streambank collapse). The METS sub-clusters shown in the lower half of Figure 3.13 (sub-clusters B), were differentiated by temporal information that was not fully captured by the Phase I hysteresis classification. Both sub-clusters are characterized by hydrographs and sedigraphs that return more completely (relative to sub-clusters A) to baseline levels. Whether used on its own or on a dataset that has been pre-classified or grouped by some other means, METS offers hydrological researchers a flexible and powerful approach for data-driven analysis of high-frequency water quality data; and the methodology may be easily adapted to different analysis objectives.

3.5.4 Challenges and Opportunities

The sparsity of hydrological events is an inherent data challenge that relies on data-driven or machine learning methods of analysis. Our study area, a typical humid and temperate watershed, experiences on average about 30 rainfall-runoff (i.e., storm) events a year. Other recent, prominent event-based studies (Wymore et al., 2019; Sherriff et al., 2016; Vaughan et al., 2017) are similarly constrained by event sizes ranging between 8 and 90 events per monitoring site. Albeit large from an environmental monitoring perspective, these relatively small sample sizes cause significant challenges for machine learning methods. The challenges are compounded when analyzing multivariate time series generated from in-situ sensors that must be kept online during extreme events and operating simultaneously. Currently, the hydrological informatics community is investing significantly in the integration and maintenance of data hubs that comprise multiple researchers across multiple

organizations such as those of the Consortium of Universities for the Advancement of Hydrological Sciences, Inc. (CUAHSI, 2019). Despite the development of new machine learning methods to address data sparsity issues, another promising approach is to generate synthetic hydrological storm events as demonstrated in this work.

METS clustering operates on delineated events and is influenced by the degree to which both time series (SSC and Q) return (or not) to base levels at the end of the event. This highlights the importance of precise event delineation in METS clustering. In hydrology, many event-based studies rely on semi-automated and somewhat subjective methods to identify the start and end of an event, particularly when handling multipeak (consecutive) events (Wymore et al., 2019; Vaughan et al., 2017; Hamshaw et al., 2018; Sherriff et al., 2016; Gellis, 2013). Automation of event delineation is another area that can benefit from advances in machine learning methods, new data hubs, and access to synthetic, pre-delineated event data.

A key challenge with any clustering method is determining the optimal number, K, of categories (e.g., the correct number of storm event types). In this work, we select K based on the inflection point of an elbow plot. However, identifying the inflection point is often subjective. This is further complicated in hydrogeological applications, where the optimal number of categories is dependent on both the research objectives as well as the geographic location. In this proof-of-concept, we made no assumptions or preconceptions about the desired number of outcome categories. However, domain experts familiar with a particular region of study may have intuitive knowledge regarding the desired number of outcomes. Varying the number of clusters in METS is relatively straightforward and not computationally intensive; thus, researchers

can easily evaluate the effect of cluster number – particularly when methods for evaluating "optimal" (e.g., the elbow method) are not definitive. Alternatively, one could replace the METS clustering algorithm with an alternative algorithm such as the density-based clustering algorithm of Ester et al. (1996), which does not require the number of clusters as an input.

The METS clustering approach is applicable to any water quality constituent or solute (e.g., nitrate, phosphorous and conductivity), which would be expected to demonstrate very different C-Q-T trajectories and resulting clusters compared to suspended sediment concentration response (Lloyd et al., 2016a; Zuecco et al., 2016). Additionally, the approach may be extended beyond a single parameter (e.g., SSC) to multiple parameters (e.g., SSC and nitrate) to explore/reveal any unknown interactions during storm events. Expansion to multiple parameters will bring interesting visualization and analysis challenges. One approach may be to visualize events as 3-D signal trajectories such as those we presented in this work.

3.6 Conclusion

The rapidly increasing volume and availability of high-frequency time series data offer considerable opportunity to analyze watershed systems at the storm event scale. In this work, we introduce the multivariate event time series (METS) approach for categorizing hydrological storm events into a limited number of clusters given data from multiple sensors deployed in the Mad River watershed in Vermont, USA. In order to validate the approach, we showed that stochastic generation of synthetic

hydrographs and concentration graphs provided a simple and effective solution to over-coming the data sparsity challenge in training machine learning algorithms on environmental data. The approach is flexible enough to be used with any water quality constituents (e.g., nitrate, phosphorous and conductivity) alone or in combination. We highlight areas for further research to expand the application of event-based analysis. Additionally, we discuss how the METS clustering can be used in tandem with a traditional hysteresis based event classification scheme. Whether used on its own or in tandem with other partitioning methods, this method offers hydrological researchers a flexible and powerful approach for analyzing high-frequency water quality data; and opens up new possibilities for interpreting emergent event behavior in watersheds.

3.7 Acknowledgements

This project was supported by the Richard Barrett Foundation and Gund Institute for Environment through a Gund Barrett Fellowship. Additional support was provided by the Vermont EPSCoR BREE Project (NSF Award OIA-1556770). We thank Dr. Jae-Gil Lee, Associate Professor at the Korea Advanced Institute of Science and Technology (KAIST), for guiding the algorithm selection, and Dr. Patrick J. Clemins of Vermont EPSCoR, for providing support in using the EPSCoR Pascal high-performance computing server for the project.

Bibliography

Aguilera, R. and Melack, J. M. (2018). Concentration-discharge responses to storm events in coastal California watersheds. *Water Resources Research*, 54(1):407–424.

Banerjee, A. and Dave, R. N. (2004). Validating clusters using the Hopkins statistic. In *Proceedings of the 2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, volume 1, pages 149–153.

Begum, N., Ulanova, L., Wang, J., and 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*, pages 49–58.

Bellman, R. (1957). *Dynamic Programming*. Dover Publications.

Bende-Michl, U., Verburg, K., and 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.

Burns, D. A., Pellerin, B. A., Miller, M. P., Capel, P. D., Tesoriero, A. J., and Duncan, J. M. (2019). Monitoring the riverine pulse: Applying high-frequency nitrate data to advance integrative understanding of biogeochemical and hydrological processes. *Wiley Interdisciplinary Reviews: Water*, page e1348.

Burt, T. P., Worrall, F., Howden, N. J. K., and Anderson, M. G. (2015). Shifts in discharge-concentration relationships as a small catchment recover from severe drought. *Hydrological Processes*, 29(4):498–507.

Chen, L., Sun, C., Wang, G., Xie, H., and Shen, Z. (2017). Event-based nonpoint source pollution prediction in a scarce data catchment. *Journal of Hydrology*, 552:13–27.

CUAHSI (2019). Consortium of universities for the advancement of hydrologic science, inc. https://www.cuahsi.org.

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., and Hexagon-ML (2018). The UCR time series classification archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Dierckx, P. (1993). *Curve and Surface Fitting with Splines*. Oxford University Press, Inc.

Dupas, R., Tavenard, R., Fovet, O., Gilliet, N., Grimaldi, C., and Gascuel-Odoux, C. (2015). Identifying seasonal patterns of phosphorus storm dynamics with dynamic

time warping. *Water Resources Research*, 51(11):8868–8882.

Ehret, U. and Zehe, E. (2011). Series distance - An intuitive metric to quantify hydrograph similarity in terms of occurrence, amplitude and timing of hydrological events. *Hydrology and Earth System Sciences*, 15(3):877–896.

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

Ewen, J. (2011). Hydrograph matching method for measuring model performance. *Journal of Hydrology*, 408(1):178 – 187.

Gellis, A. (2013). Factors influencing storm-generated suspended-sediment concentrations and loads in four basins of contrasting land use, humid-tropical Puerto Rico. *CATENA*, 104:39 – 57.

Hamshaw, S., M. Dewoolkar, M., W. Schroth, A., Wemple, B., and M. Rizzo, D. (2018). A new machine-learning approach for classifying hysteresis in suspended-sediment discharge relationships using high-frequency monitoring data. *Water Resources Research*, 54:4040–4058.

Javed, A. (2019a). Dynamic Time Warping. https://github.com/ali-javed/dynamic-time-warping.

Javed, A. (2019b). K-medoids for multivariate time series clustering. https://github.com/ali-javed/Multivariate-Kmedoids.

Keesstra, S. D., Davis, J., Masselink, R. H., CasalĂ, J., Peeters, E. T. H. M., and Dijksma, R. (2019). Coupling hysteresis analysis with sediment and hydrological connectivity in three agricultural catchments in Navarre, Spain. *Journal of Soils and Sediments*, 19(3):1598–1612.

Latecki, L. J., Megalooikonomou, V., Qiang Wang, Lakaemper, R., Ratanamahatana, C. A., and Keogh, E. (2005). Partial elastic matching of time series. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 4 pp.–.

Latecki, L. J., Megalooikonomou, V., Wang, Q., Lakaemper, R., Ratanamahatana, C. A., and Keogh, E. (2005). Elastic partial matching of time series. In *Knowledge Discovery in Databases*, pages 577–584.

Lloyd, C., Freer, J., Johnes, P., and Collins, A. (2016a). Using hysteresis analysis of high-resolution water quality monitoring data, including uncertainty, to infer controls on nutrient and sediment transfer in catchments. *Science of The Total Environment*, 543, Part A:388 – 404.

Lloyd, C. E. M., Freer, J. E., Johnes, P. J., and Collins, A. L. (2016b). Technical Note: Testing an improved index for analysing storm discharge–concentration hysteresis. *Hydrology and Earth System Sciences*, 20(2):625–632.

Malutta, S., Kobiyama, M., Chaffe, P. L. B., and BonumĂĄ, N. B. (2020). Hysteresis analysis to quantify and qualify the sediment dynamics: state of the art. *Water Science and Technology*. wst2020279.

Mather, A. L. and Johnson, R. L. (2014). Quantitative characterization of stream turbidity-discharge behavior using event loop shape modeling and power law parameter decorrelation. *Water Resources Research*, 50(10):7766–7779.

Mather, A. L. and 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.

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

Nadal-Romero, E., RegĂźĂŠs, D., and Latron, J. (2008). Relationships among rainfall, runoff, and suspended sediment in a small catchment with badlands. *CATENA*, 74(2):127 – 136.

Onderka, M., Krein, A., Wrede, S., Martinez-Carreras, N., and Hoffmann, L. (2012). Dynamics of storm-driven suspended sediments in a headwater catchment described by multivariable modeling. *Journal of Soils and Sediments*, 12(4):620–635.

Paparrizos, J. and Gravano, L. (2016). K-shape: Efficient and accurate clustering of time series. *SIGMOD Record*, 45(1):69–76.

Paparrizos, J. and Gravano, L. (2017). Fast and accurate time-series clustering. *ACM Transactions on Database Systems*, 42(2):8:1–8:49.

PRISM (2019). PRISM climate group. http://prism.oregonstate.edu. Last accessed on March 16, 2019.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and 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*, pages 262–f–270.

Ratanamahatana, C. A. and Keogh, E. (2004). Everything you know about Dynamic Time Warping is wrong. In *Proceedings of the 3rd Workshop on Mining Temporal*

and Sequential Data. Citeseer.

Rose, L. A., Karwan, D. L., and Godsey, S. E. (2018). Concentration-discharge relationships describe solute and sediment mobilization, reaction, and transport at event and longer timescales. *Hydrological Processes*, 32(18):2829–2844.

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

Savitzky, A. and Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639.

Seeger, M., Errea, M.-P., Begueria, S., Arnaez, J., Marti, C., and Garcia-Ruiz, J. (2004). Catchment soil moisture and rainfall characteristics as determinant factors for discharge/suspended sediment hysteretic loops in a small headwater catchment in the spanish pyrenees. *Journal of Hydrology*, 288(3):299–311.

Sherriff, S. C., Rowan, J. S., Fenton, O., Jordan, P., Melland, A. R., Mellander, P.-E., and hUallachĂĄin, D. O. (2016). Storm event suspended sediment-discharge hysteresis and controls in agricultural watersheds: Implications for watershed scale sediment management. *Environmental Science & Technology*, 50(4):1769–1778.

Shokoohi-Yekta, M. and Keogh, E. J. (2015). On the non-trivial generalization of Dynamic Time Warping to the multi-dimensional case. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 289–297.

Stryker, J., Wemple, B., and Bomblies, A. (2017). Modeling sediment mobilization using a distributed hydrological model coupled with a bank stability model. *Water Resources Research*, 53(3):2051–2073.

Vaughan, M. C. H., Bowden, W. B., Shanley, J. B., Vermilyea, A., Sleeper, R., Gold, A. J., Pradhanang, S. M., Inamdar, S. P., Levia, D. F., Andres, A. S., and et al. (2017). High-frequency dissolved organic carbon and nitrate measurements reveal differences in storm hysteresis and loading in relation to land cover and seasonality: high-resolution doc and nitrate dynamics. *Water Resources Research*, 53:5345–5363.

Wemple, B. C., Clark, G. E., Ross, D. S., and Rizzo, D. M. (2017). Identifying the spatial pattern and importance of hydro-geomorphic drainage impairments on unpaved roads in the northeastern usa. *Earth Surface Processes and Landforms*, 42(11):1652–1665.

Wendi, D., Merz, B., and Marwan, N. (2019). Assessing hydrograph similarity and rare runoff dynamics by cross recurrence plots. *Water Resources Research*, 55(6):4704–4726.

Williams, G. P. (1989). Sediment concentration versus water discharge during single hydrologic events in rivers. *Journal of Hydrology*, 111(1):89–106.

Williams, M. R., Livingston, S. J., Penn, C. J., Smith, D. R., King, K. W., and Huang, C.-h. (2018). Controls of event-based nutrient transport within nested headwater agricultural watersheds of the western Lake Erie basin. *Journal of Hydrology*, 559:749–761.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., and Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37.

Wymore, A. S., Leon, M. C., Shanley, J. B., and McDowell, W. H. (2019). Hysteretic response of solutes and turbidity at the event scale across forested tropical montane watersheds. *Frontiers in Earth Science*, 7:126.

Zuecco, G., Penna, D., Borga, M., and van Meerveld, H. J. (2016). A versatile index to characterize hysteresis between hydrological variables at the runoff event timescale. *Hydrological Processes*, 30(9):1449–1466.

3.8 Supporting Information

This supporting information contains tables and figures to provide additional information on the following aspects of the study:

1. Table S1: Study watershed characteristics.
2. Figure S1: Matrix representation of multivariate time series.
3. Algorithm S1: K-medoids algorithm for hydrological event clustering.
4. Algorithm S2: Dynamic time warping algorithm for calculating the distance between two time series.
5. Table S2: Default parameter settings for synthetic hydrograph and concentration-graph generator.

6. Table S3: Result of post-hoc Tukey HSD test for all pairwise comparisons of antecedent conditions metrics.

7. Table S4: Result of post-hoc Tukey HSD test for all pairwise comparisons of rainfall characteristics metrics.

8. Table S5: Result of post-hoc Tukey HSD test for all pairwise comparisons of streamflow and sediment characteristics metrics.

9. Figure S2: SSE for varying number of clusters for Mad River dataset and Expanded dataset.

10. Table S6: Distribution of hysteresis loop classes over METS cluster 5 (when K= 9) in the expanded dataset(n= 56).

11. Figure S3: Three storm events closest to the centroid of the four extended dataset tandem clockwise hysteresis sub-clusters (K= 4, N= 496).

12. Figure S4: Three storm events closest to the centroid of the four extended dataset tandem counter clockwise hysteresis sub-clusters (K= 2, N= 90).

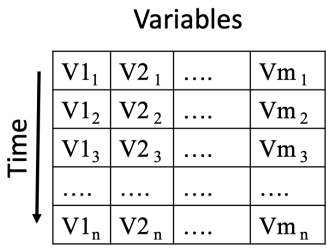


Figure S1: A matrix representation of multivariate time series (m variables, n time steps); a column for each variable and a row for variable value at each time step.

Table S1: Study watershed characteristics.

Characteristic	**Freeman Brook**	**Folsom Brook**	**Mill Brook**	**High Bridge Brook**	**Shepard Brook**	**Mad River**	**Allen Brook**	**Hungerford Brook**	**Wade Brook**
Area (km^2)	17.0	18.2	49.2	8.6	44.6	344.0	25.5	16.7	48.1
Minimum elevation (m)	266	229	216	225	195	140	61	320	33
Maximum elevation (m)	860	886	1114	796	1117	1245	351	981	354
Elevation range (m)	594	657	898	571	923	1105	290	661	321
Stream order	4th	4th	4th	3rd	4th	5th	3rd	3rd	5th
Drainage density (km/km^2)	1.95	1.77	2.16	2.45	2.38	0.97	1.81	1.57	2.28
% Forested land	76.2	77.6	89.2	66.7	92.2	85.5	39.3	95.1	40.5
% Developed land	8.3	12.7	1.5	16.6	1.0	4.7	26.5	0.8	7.9
% Agricultural land	14.6	8.8	7.0	15.5	5.6	8.0	28.6	0.6	44.8
% Other land	1.7	0.7	0.8	2.1	1.1	1.1	5.6	3.5	6.8

Algorithm K-medoids

Input: storm events (i.e., their multivariate time series representations); number k of clusters to be generated.

Output: k clusters generated from the events.

Procedure

Randomly select k events as medoids from the input events.

while *termination criteria are not met* **do**
  // Termination condition can be convergence of medoids or maximum allowed iterations.
  Phase 1: Assign each event to its closest medoid.
  Phase 2: From each cluster consisting of the medoid and events assigned to it, select an event that gives the smallest sum of distances to all the other events in the cluster and make the selected event a new medoid.
end
Return each cluster, consisting of a medoid and all events assigned to it.

Algorithm S1: K-medoids algorithm for hydrological event clustering.

Algorithm DTW

Input: $T1$, $T2$: time series, W: warping window size

Output: distance between $T1$ and $T2$

Procedure

Let a and b be the lengths of $T1$ and $T2$, respectively.
Let m be the number of variables in $T1$ and $T2$, respectively.
Create a distance matrix D of size $a \times b$ and initialize all matrix elements to ∞.
$D[0,0] := 0$. // Initialize the first entry in D.
$i := 1$. $j = 1$. // Initialize the index of a warping path between $T1$ and $T2$.
while $i \leq a$ and $j \leq b$ **do**
    Calculate the squared Euclidean distance, $\sum_{c=1}^{m}(t1_i^c - t2_j^c)^2$, between the ith item in $T1$ and each of the jth item in $T2$ within the range of $j = [i - W, i + W]$.
    Update $D[i,j]$ to $\sum_{c=1}^{m}(t1_i^c - t2_j^c)^2 +$ $\min\{D[i-1,j], D[i,j-1], D[i-1,j-1]\}$.
    increase i by 1.
end
return $\sqrt{D[a,b]}$.

Algorithm S2: Dynamic time warping algorithm for calculating the distance between two time series.

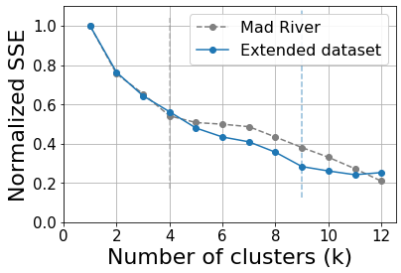


Figure S2: SSE for varying number of clusters for Mad River dataset and Expanded dataset.

Table S2: Default parameter settings for synthetic hydrograph and concentration-graph generator.

Hydrograph					
Type	**Duration of peak**	**Time to peak**	**Delay**	**Recess**	**Initial Baseflow**
Flashy - early peak return to baseflow	0.4	0.5	0	0.1	0
Flashy - early peak incomplete return to baseflow	0.4	0.5	0	0.4	0
Early peak slow return to baseflow	0.8	0.2	0	0.1	0
Early peak incomplete return to baseflow	0.8	0.2	0	0.4	0
Mid-peak return to baseflow	0.8	0.5	0	0.1	0
Mid-peak incomplete return to baseflow	0.8	0.5	0	0.4	0
Delayed rise to mid-peak return to baseflow	0.8	0.5	0.2	0.1	0.1
Delayed rise to mid-peak incomplete return to baseflow	0.8	0.5	0.2	0.4	0.1
Concentration-graph					
Type	**Duration**	**Time to peak**	**Onset**	**Recess**	**Storm flow**
Early peak	0.5	0.5	0	0	0
Late peak	0.5	0.5	0.5	0	0

Table S3: Result of post-hoc Tukey HSD test for all pairwise comparisons of antecedent conditions metrics. Within each classification scheme if two classes/clusters do not share a letter the mean metric value is significantly different (alpha = 0.05).

Antecedent conditions						
Metric	T_{LASTP}	$A3P$	$A14P$	$SM_{SHALLOW}$	SM_{DEEP}	BF_{NORM}
METS clusters						
cluster 1	a	a	a	a	a	a
cluster 2	a	b	b	a	a	b
cluster 3	a	b c	b	a	a	a b
cluster 4	a	c	b	a	a	b
Hysteresis classes						
Class I	a	a b	a	a	a	a
Class II	a	a	a	a	a	a
Class III	a	b	a	a	a	a
Class IV	a	a b	a	a	a	a
Class V	a	a b	a	a	a	a
Complex	a	a b	a	a	a	a

Table S4: Result of post-hoc Tukey HSD test for all pairwise comparisons of rainfall characteristics metrics. Within each classification scheme if two classes/clusters do not share a letter the mean metric value is significantly different (alpha = 0.05).

Rainfall characteristics				
Metric	P	P_{MAX}	D_P	T_{PSSC}
METS clusters				
cluster 1	a	a	a	a
cluster 2	b	b	a	b
cluster 3	b	b	a	a
cluster 4	a	c	b	b
Hysteresis classes				
Class I	a b	a	a b	a b
Class II	a	a	a	c
Class III	b	a	a	d
Class IV	a b	a	a b	a
Class V	a b	a	a b	a b
Complex	a b	a	b	b d

Table S5: Result of post-hoc Tukey HSD test for all pairwise comparisons of streamflow and sediment characteristics metrics. Within each classification scheme if two classes/clusters do not share a letter the event metric value is significantly different (alpha = 0.05).

Streamflow and sediment characteristics					
Metric	BL	Q_{NORM}	$Log(Q_{NORM})$	SSL_{NORM}	$FLUX_{NORM}$
METS clusters					
cluster 1	a	a	a	a	a
cluster 2	b	b	b	a	a
cluster 3	a c	a b	b	a	a
cluster 4	b c	a b	a	a	a
Hysteresis classes					
Class I	a b	a	a b	a	a
Class II	c	a	a	a	a
Class III	a b	a	b	a	b
Class IV	a c	a	a b	a	a
Class V	a b	a	a b	a	a
Complex	b	a	a b	a	a

Hysteresis class	**Count**
I - Linear (Counter-clockwise)	7
II - Clockwise	0
III - Counter-clockwise	30
IV - Linear then clockwise	0
V - Figure eight	9
Complex (Counter-clockwise)	10
Total	56

Table S6: Distribution of hysteresis loop classes over METS cluster 5 (when $K = 9$) in the expanded dataset ($n = 56$).

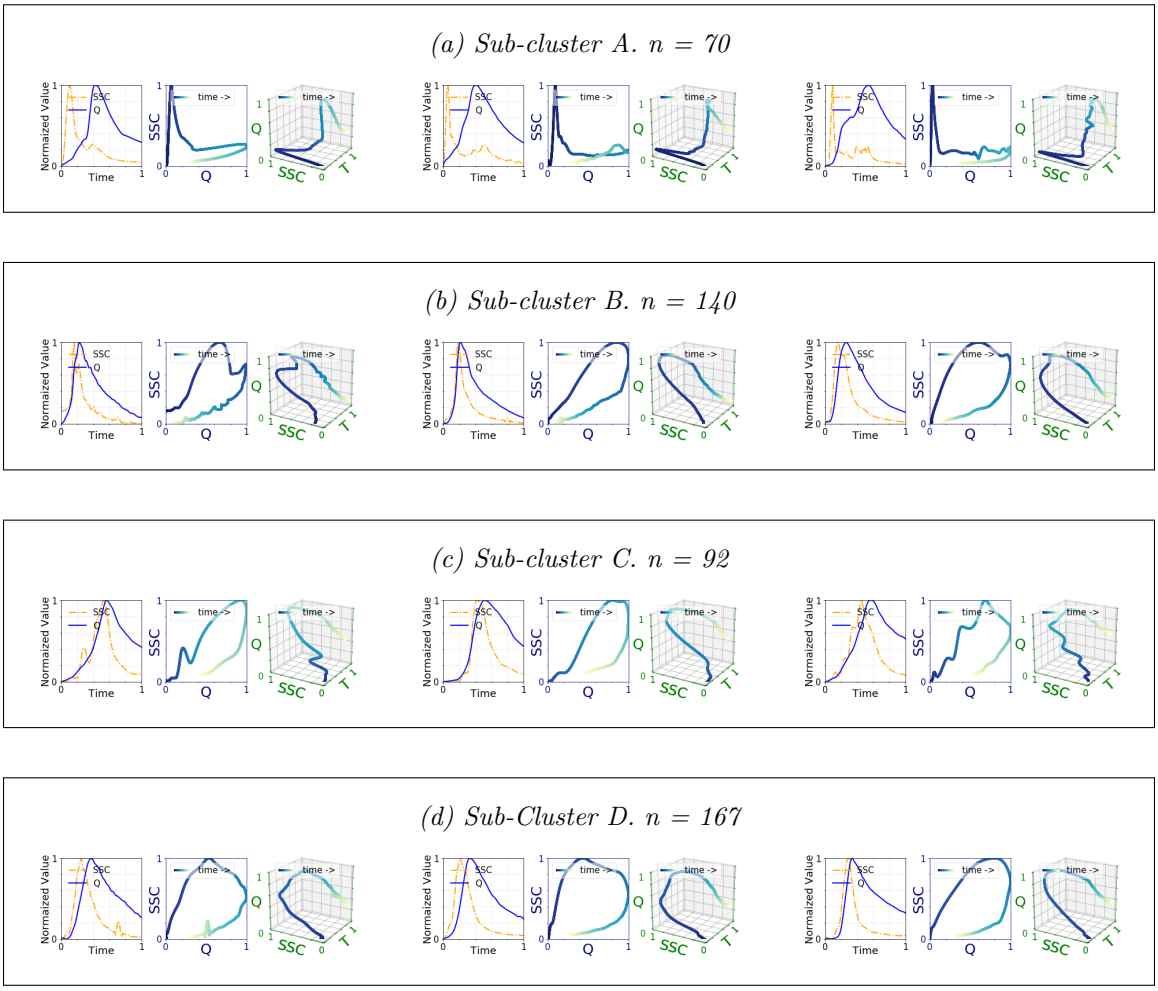


Figure S3: Three storm events closest to the centroid of the four extended dataset tandem clockwise hysteresis sub-clusters ($K = 4$, $N = 496$) — (a) cluster 1 events have sedigraph peaks that occur well before the hydrographs resulting in an "L" shaped loop, (b) cluster 2 have quickly rising hydrographs and sedigraphs, (c) cluster 3 have slow rising hydrographs and sedigraphs, and (d) cluster 4 have sedigraphs that peak before the hydrographs resulting in broad clockwise loops.

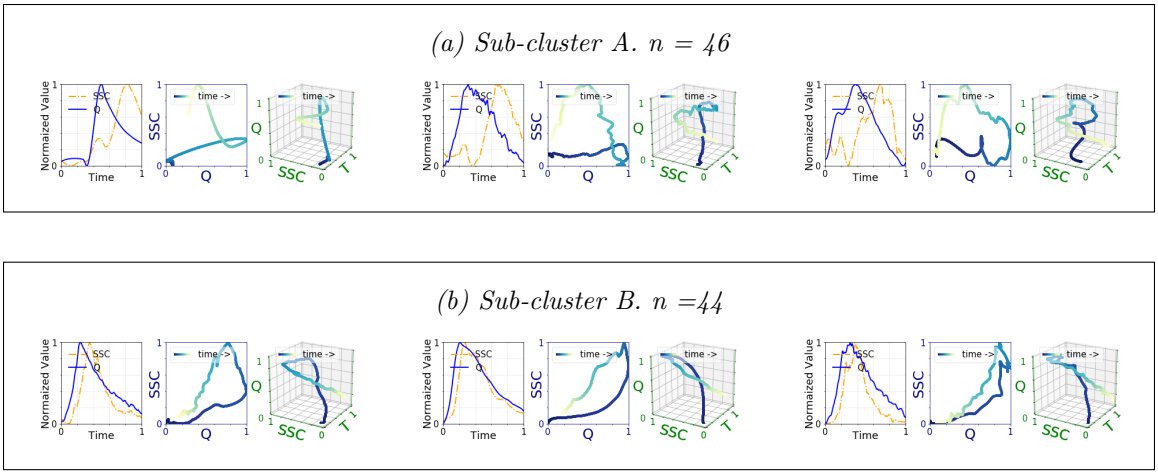


Figure S4: Three storm events closest to the centroid of the four extended dataset tandem counter clockwise hysteresis sub-clusters ($K = 2$, $N = 90$) — (a) cluster 1 events have sedigraph peaks that occur well after the hydrographs resulting in an approximate mirror image of "L" shaped loop and (b) cluster 2 events have sedigraph peaks that occur slightly after the hydrograph peaks.

Chapter 4

SOMTimeS: Self Organizing Maps for Time Series Clustering and its Application to Serious Illness Conversations

ABSTRACT

There is an increasing demand for scalable algorithms capable of clustering and analyzing large time series datasets. The Kohonen self-organizing map (SOM) is a type of unsupervised artificial neural network used for visualizing and clustering complex data, reducing the dimensionality of data, and selecting influential features. Like all clustering methods, the SOM requires a means of measuring similarity between input observations (in this work time series data). Dynamic time warping (DTW) is one such method, and a top performer given that it is resilient to the distortions in time series alignment. Despite its prior use in clustering methods, including the SOM algorithm, DTW is limited in practice because this resilience comes at a high computational cost when clustering large amounts of data associated with real applications; DTW is quadratic in runtime complexity with the length of the time series data. To address this, we present a new DTW-based clustering method, called SOMTimeS (a Self-Organizing Map for TIME Series) that uses DTW, yet scales better and runs faster than competing DTW-based clustering methods. The computational performance of SOMTimeS stems from its ability to prune unnecessary DTW computations during the SOM's unsupervised learning (i.e., training) phase. We evaluated the performance accuracy and scalability on 112 benchmark time series datasets from the University of California, Riverside classification archive. SOMTimeS clustered data with state-of-the-art accuracy.

That is - it matched and in some cases exceed the performance accuracy of the two top-performing DTW-based clustering methods, namely K-means and TADPole. However, the computational pruning demonstrated an empirical runtime complexity (with respect to the number and length of time series) that is 1.6x and 10x times faster than K-means and TADPole, respectively. SOMTimeS also inherits the outstanding visualization ability of SOM. In this regard, we apply SOMTimeS to a complex time series of conversational features extracted from natural language conversation data collected as a part of a large healthcare cohort study of patient-family-clinician serious illness discussions.

4.1 Introduction

By 2025, it is estimated that more than four hundred fifty exabytes of data will be collected and stored every day (WorldEconomicForum, 2019). Much of that data will be collected continuously and represent phenomena that change over time. We propose that fully understanding the meaning of these data will often require complexity scientists to model them as time series. Examples include data collected by sensors (CRS, 2020; Evans, 2011), every day natural language (e.g., Bentley et al., 2018; Ross et al., 2020; Reagan et al., 2016; Chu et al., 2017), biomonitors (Gharehbaghi and Linden, 2018), waterflow, barometric pressure and other routine environmental condition meters (e.g., Hamami and Dahlan, 2020; Javed et al., 2020a; Ewen, 2011), social media interactions (e.g., De Bie et al., 2016; Javed and Lee, 2018, 2016, 2017), and hourly financial data reported by fluctuating world stock and currency markets (Lasfer et al., 2013). In response to the increasing amounts of time-oriented data available to analysts, the applications of time-series modeling are growing rapidly (e.g., Minaudo et al., 2017; Dupas et al., 2015; Mather and Johnson, 2015; Bende-Michl et al., 2013; Iorio et al., 2018; Gupta and Chatterjee, 2018; Pirim et al., 2012; Souto et al., 2008; Flanagan et al., 2017).

Time series modeling is computationally "expensive" in terms of processing power and speed of analysis. Indeed, as the numbers of observations or measurement dimensions for each observation increase, the relative efficiency of time series modeling diminishes, creating an exponential deterioration in computational speed. Under

conditions where computing power is in excess or when result generating speed is unimportant, these challenges would be less pressing. However, these conditions are rarely met currently, and the accelerating rate of data collection promises to continue outpacing the computational infrastructure available to most analysts.

In this work, we present SOMTimeS (Self-organizing Map for Time Series), a computationally efficient and fast time series clustering algorithm for application to large time series datasets. The computational efficiency of SOMTimeS is attributed to the pruning of unnecessary DTW computations during the SOM training phase. When assessed using 112 time series datasets belonging to different domains from the University of California, Riverside (UCR) classification archive, 43% of the needed DTW computations were pruned. Empirically the pruning rate increased proportional to the increase in DTW computation time as the length of time series increased. As a result, the algorithm scales better with increasing data, making SOMTimeS, to the best of our knowledge, the fastest DTW-based clustering algorithm to date.

SOMTimeS also inherits the outstanding visualization ability of SOM. In order to explore the potential utility of SOMTimeS to novel problems of high complexity and natural sequential ordering of data, we evaluated its performance when applied to the science of doctor-family-patient conversations in high emotion settings. Understanding and improving serious illness communication is a national priority for 21st century healthcare, but our existing methods for measuring and analyzing data is cumbersome, human intensive, and far too slow to be relevant for large epidemiological studies, communication training or time-sensitive reporting. Here, we use data from an existing multi-site epidemiological study of healthcare serious illness conversations

as one example of how efficient computational methods can add to the science of healthcare communication.

The remainder of this paper is organized as follows. Section 4.2 provides background information on SOMs and DTW. Section 4.3 presents the SOMTimeS algorithm. Section 4.4 and Section 4.5 evaluate the results of SOMTimeS clustering algorithm on UCR benchmark datasets and serious illness discussions, respectively. Section 4.6 discusses the results. Section 4.7 concludes the paper and suggests future work.

4.2 Background

Similar to the work by Silva and Henriques (2020), Li et al. (2020), Parshutin and Kuleshova (2008) and, Somervuo and Kohonen (1999), SOMTimeS combines an artificial neural network known as the Kohonen SOM with the robust DTW-based distance measure. While the Kohonen SOM (see details in Section 4.2.1) is linearly scalable with respect to the number of input data, it often performs hundreds of passes (i.e., epochs) when self-organizing or clustering the training data. Each epoch requires $n \times M$ distance calculations, where n is the number of observations and M is the number of nodes in the network map. This large number of distance calculations is problematic, particularly when the distance measure is computationally expensive, as is the case with DTW (see Section 4.2.2).

DTW, originally introduced in 1970s for speech recognition (Sakoe and Chiba, 1978), continues to be one of the more robust, top performing, and consistently

chosen learning algorithms for time series data (Xi et al., 2006; Ding et al., 2008; Paparrizos and Gravano, 2016, 2017; Begum et al., 2015; Javed et al., 2020b). Its ability to shift, stretch, and squeeze portions of the time series helps address challenges inherent to time series data (e.g., aligning the shapes of conversational story arcs). Unfortunately, the ability to distort the temporal dimension comes with increased computational overhead, which has hindered its use in practical applications involving large datasets or long time series clustering (Javed et al., 2020b; Zhu et al., 2012). The first subquadratic-time algorithm ($O(m^2/\log\log m)$) for DTW computation was proposed by Gold and Sharir (2018), which is still more computationally expensive in comparison to the simpler Euclidean distance ($O(m)$).

To address the computational cost, several studies have presented approximate solutions (Zhu et al., 2012; Salvador and Chan, 2007a; Al-Naymat et al., 2009). To the best of our knowledge, TADPole by Begum et al. (2015) is the only algorithm (see supplementary material Section 4.8.1) that speeds up the DTW computation without using an approximation. It does so by using a bounding mechanism to prune the expensive DTW calculations. Yet, when coupled with the clustering algorithm (i.e., Density Peaks of Rodriguez and Laio (2014)), it still scales quadratically. Thus, even after decades of research (Zhu et al., 2012; Begum et al., 2015; Lou et al., 2015; Salvador and Chan, 2007b; Wu and Keogh, 2020), the almost quadratic time complexity of DTW-based clustering still poses a challenge when clustering time series in practice.

4.2.1 Self Organizing Maps

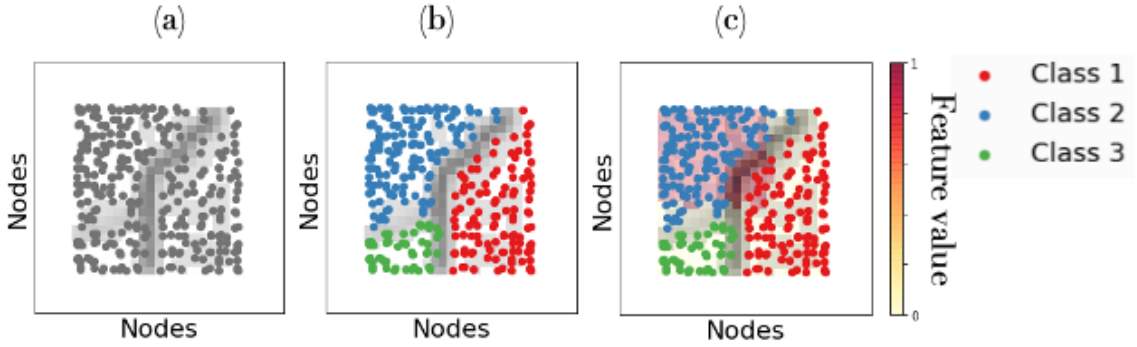


Figure 4.1: Self-organizing maps used for clustering and visualizing times series observations from the UCR archive dataset — InsectEPGRegularTrain; the self-organized data are shown with (a) a unified distance matrix, (b) color-coded clusters, and (c) a single input variable (or feature value) in the background.

The Kohonen Self-Organizing Map (Kohonen et al., 2001; Kohonen, 2013) may be used for either clustering or classifying observations, and has advantages when visualizing complex, nonlinear data (Alvarez-Guerra et al., 2008; Eshghi et al., 2011). Additionally, it has been shown to outperform other parametric methods on datasets containing outliers or high variance (Mangiameli et al., 1996). Similar to methods such as logistic regression and principal component analysis, SOMs may be used for feature selection, as well as mapping input data from a high-dimensional space to a lower-dimensional space (typically a two-dimensional mesh or lattice); one such example is shown in Figure 4.1. The SOM clustering results using input data from one of the datasets in the UCR classification archive (InsectEPGRegularTrain) have been self-organized onto a 2-D mesh. Each gray node represents a time series (i.e., temporal pattern or arc). The self-organized observations may be plotted with what is known as a unified distance matrix or U-matrix (Ultsch, 1993). The latter is obtained

by calculating the average difference between the weights of adjacent nodes in the trained SOM, and then plotting these values (in a gray scale of Figure 4.1) on the trained 2-D mesh. Darker shading represents higher U-matrix values (larger average distance between observations). In this manner, the U-matrix can help assess the quality and the number of clusters. For example, see the U-matrix of Figure 4.1(b), which separates the observations into three clusters that may be color-coded or labeled (should labels exist). Finally, any information, input features, or metadata associated with the observations may be visualized (red shading of Figure 4.1(c)) in the same 2-D space in order to explore associations and the importance of individual input features with the clustered results. The ability to visualize individual input features in the same space as the clustered observations (known as component planes) makes the SOM a powerful tool for data analysis and feature selection.

4.2.2 Dynamic time warping

DTW is recognized as one of the most accurate similarity measures for time series data (Paparrizos and Gravano, 2017; Rakthanmanon et al., 2012; Johnpaul et al., 2020). While the most common measure, Euclidean distance, uses a one-to-one alignment between two time series (e.g., labeled candidate and query in Figure 4.2(a)), DTW employs a one-to-many alignment that warps the time dimension (see Figure 4.2(b)) in order to minimize the sum of distances between time series samples. As such, DTW can optimize alignment both globally (by shifting the entire time series left or right) and locally (by stretching or squeezing portions of the time

series). The optimal alignment should adhere to three rules:

1. Each point in the query time series must be aligned with one or more points from candidate time series, and vice versa.

2. The first and last points of the query time series and a candidate time series must align with each other.

3. No cross-alignment is allowed; that is, the aligned time series indices must increase monotonically.

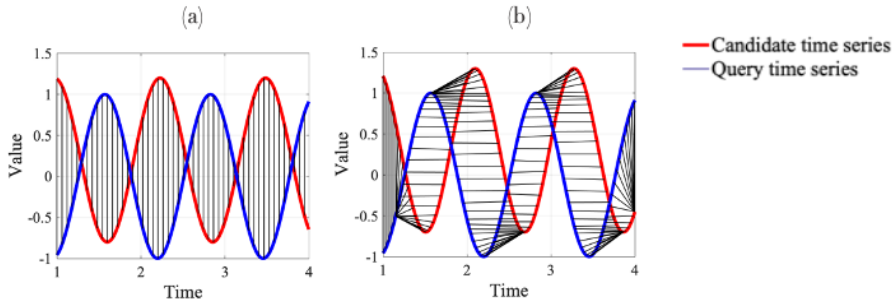


Figure 4.2: Alignment between two times series for calculating (a) Euclidean distance and (b) DTW distance.

DTW is often restricted to aligning points only within a moving window of a fixed size to improve accuracy and reduce computational cost. The window size may be optimized using supervised learning on training data, but for clustering where supervised learning is not possible, a window size amounting to 10% of the observation data is usually considered adequate (Ratanamahatana and Keogh, 2004).

Upper and lower bounds of DTW distance

SOMTimeS uses distance bounding to prune the DTW calculations performed during the SOM unsupervised learning. This distance bounding involves finding a tight upper

and lower bound. Because DTW is designed to find a mapping that minimizes the sum of the point-to-point distances between two time series, that mapping can never result in a distance that is greater than the sum of point-to-point Euclidean distance. Hence, finding the tight upper bound is straight forward – it is the Euclidean distance (Keogh, 2002). To find the lower bound, we use the LB_Keogh method (Keogh and Kasetty, 2003), which is commonly used in similarity searches (Keogh and Kasetty, 2003; Ratanamahatana et al., 2005; Li Wei et al., 2005) and clustering (Begum et al., 2015). The method comprises two steps (see Figure 4.3a and Figure 4.3b). Given a fixed

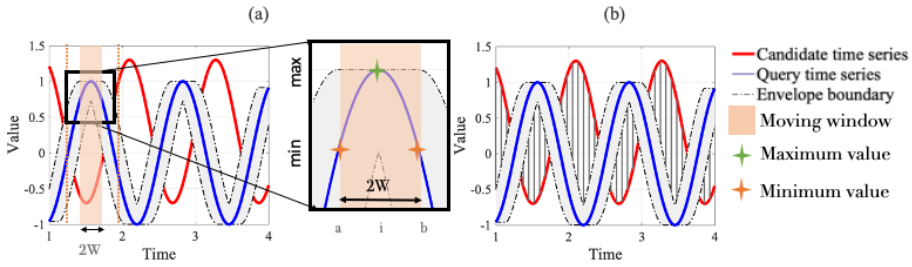


Figure 4.3: Two steps of calculating the L_Keogh tight lower bound for DTW in linear time: (a) determine the envelope around a query time series, and (b) sum the point to point distance shown in grey lines between the envelope and a candidate time series as LB_Keogh (Equation 4.2).

DTW window size, W, one of the two time series (called the query time series, Q) is bounded by an envelope having an upper (U_i) and lower boundary (L_i) calculated, respectively as:

$$U_i = \max(q_a, ..., q_i, ..., q_b)$$
$$L_i = \min(q_a, ..., q_i, ..., q_b) \tag{4.1}$$

where $a = i - W$, and $b = i + W$ (see Figure 4.3a). In the second step, the LB_Keogh lower bound is calculated as the sum of Euclidean distance between the candidate time series and the envelope boundaries (see vertical lines of Figure 4.3b). Equation 4.2

shows the formula for calculating LB_Keogh:

$$\text{LB\_Keogh} = \sqrt{\sum_{i=1}^{m} \begin{cases} (t_i - U_i)^2, & \text{if } t_i > U_i \\ (t_i - L_i)^2, & \text{if } t_i < L_i \\ 0, & \text{otherwise} \end{cases}} \tag{4.2}$$

where t_i, U_i, and L_i are the values of a candidate time series, the upper and lower envelope boundary, respectively, at time step i.

4.3 The SOMTimeS Algorithm

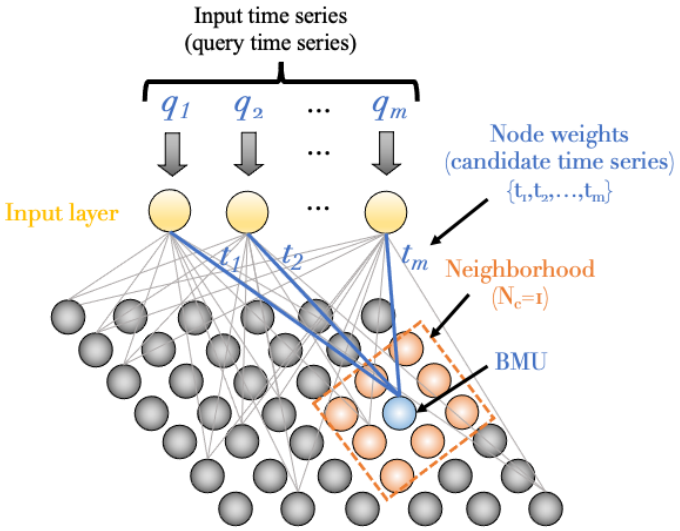


Figure 4.4: Schematic of the Kohonen Self-Organizing Map (after Kohnen, 2001) showing weights (candidate time series) of the best matching unit (BMU) in blue surrounded by a user-specified neighborhood (N_c).

SOMTimeS is a variant of the SOM (see Algorithm 4.1), where each input observation (i.e., query time series) is compared with the weights (i.e., candidate time

series) associated with each node in the 2-D mesh (see Figure 4.4). During training, the comparison (or distance calculation) between these two time series is performed to identify the SOM node whose weights are most similar to a given input time series; this node is identified as the "best matching unit (BMU)". Once the nodal weights (candidate time series) of the BMU have been identified, these weights (and those of the neighborhood nodes) are updated to more closely match the query time series. This same process is performed for all query time series in the dataset – defined as one epoch. While iterating through some user-defined fixed number of epochs, both the neighborhood size and the magnitude of change to nodal weights are incrementally reduced. This allows the SOM to self-organized or converge to a solution (stable map of clustered nodes), where the set of weights associated with these self-organized nodes now approximate the input time series (i.e., observed data). In SOMTimeS, the distance calculation is done using DTW with bounding, which helps prune the number of DTW calculations required to identify the BMU.

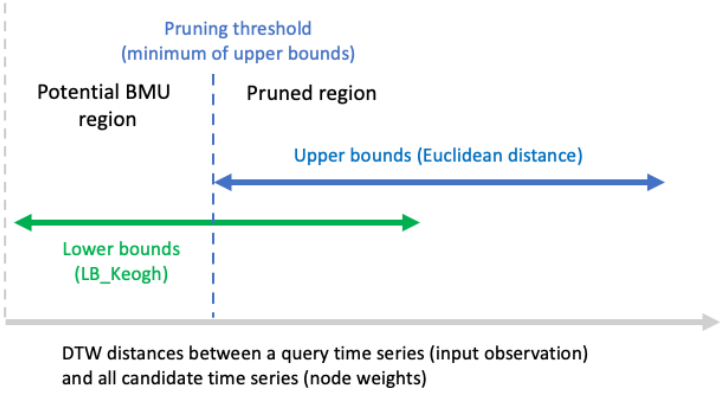


Figure 4.5: Identification of a qualification region in SOMTimeS.

The pruning is performed in two steps. First, an upper bound (i.e., Euclidean distance) is calculated between the input observation and each weight vector

associated with the SOM nodes (Line 9 of Algorithm 4.1). The minimum of these upper bounds is set as the pruning threshold (see dotted line in Figure 4.5). Next, for each SOM node we calculate a lower bound (i.e., LB_Keogh; see Line 10). If the calculated lower bound is greater than the pruning threshold, the respective node is pruned from being the BMU. If the lower bound is less than the pruning threshold, then that SOM node lies in what we call the potential BMU region (see Figure 4.5, and Line 11). As a result, the more expensive DTW calculations are performed only for the nodes in this potential BMU region; the one with the minimum summed distance is the BMU.

After identifying the BMUs for each input time series, the BMU weights, as well as the weights attached to nodes in some neighborhood of the BMUs, are updated to more closely match the respective input time series using a traditional learning algorithm based on gradient descent (Line 15 of Algorithm 4.1). Both the learning rate and the neighborhood size are reduced (see lines 16 and 17) over each epoch until the nodes have self-organized (i.e., algorithm has converged). In this work, unless otherwise stated, SOMTimeS is trained for 100 epochs. To further reduce the SOM execution time, the set of input time series (i.e., set of query time series) may be partitioned in a manner similar to Wu et al. (1991), Obermayer et al. (1990) and Lawrence et al. (1999) for parallel processing (see Line 5). We should also note that after convergence, SOMTimeS may be used to *classify* observations into a given number of clusters. This is done by setting the mesh size equal to k (desired number of clusters), and using the weights of the BMUs for direct cluster assignment.

Algorithm SOMTimeS

Input: a set $\mathcal{S}$ of query time series $\{Q_1, Q_2, ..., Q_n\}$, *epochs*: number of *epochs*, W: warping window size

Assumption: Similarity between two observation is the DTW distance between their time series.

Output: Best Matching Units

Procedure

mesh_size $M := 5 \times \sqrt{n}$ // n = the number of time series in $\mathcal{S}$

Create and randomly initialize a mesh of Nodes[$\sqrt{M}$,$\sqrt{M}$], where the weights of each node are a randomly-generated time series (candidate time series) of length equal to the query time series.

neighborhood size $N_c := \sqrt{M}/2$

learning rate $r := 0.9$

Split $\mathcal{S}$ into subsets of equal size, $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_c$, where c is the number of available CPU cores in the machine.

for *each epoch* p **do**

  for *each split* $\mathcal{S}_i$ *(*$i = 1, 2, ..., c$*) assigned to the core* i *in parallel* **do**

    for *each input time series* $Q_{i_j} (j = 1, 2, ..., n/c)$ *in* $\mathcal{S}_i$ **do**

      upper bounds:= Euclidean distances between Q_{i_j} and weights of each node.

      lower bounds:= LB_Keogh between Q_{i_j} and weights of each node using the W.

      `// Prune the set of all nodes to the set of qualified nodes.`

      $Qualified$:= Set of nodes whose weights have a lower bound with $Q_{i_j} \leq$ min(upper bounds).

      Best matching unit:= Compute DTW distance between Q_{i_j} and weights of nodes in $Qualified$. The best matching unit (BMU) is the node whose weights are most similar to Q_{i_j}.

    end

  end

  for *each time series* $Q_i (i = 1, 2, ..., n)$ **do**

    Update the node weights of the BMU (and its neighborhood) identified for Q_i using a gradient descent based on learning rate, to more closely match Q_i.

  end

  `// Update the neighborhood size and the learning rate in SOM.`

  $N_c := \sqrt{M}/2 \times (1 - p/\#epochs)$

  $r := 0.9 \times (1 - p/\#epochs)$

end

return Best matching units

Algorithm 4.1: SOMTimeS algorithm

4.4 Performance Evaluations

The UCR time series classification archive (Dau et al., 2018), with thousands of citations and downloads, is arguably the most popular archive for benchmarking time series clustering algorithms. The archive was bornout of frustration, with studies on clustering and classification reporting error rates on a single time series dataset, and then implying that the results would generalize to other datasets. At the time of this writing, the archive has 128 datasets comprising a variety of synthetic, real, raw and pre-processed time series data, and has been used extensively for benchmarking the performance of clustering algorithms (e.g., Paparrizos and Gravano, 2016, 2017; Begum et al., 2015; Javed et al., 2020b; Zhu et al., 2012). For the evaluation of SOMTimeS, we excluded sixteen of the archive datasets because they contained only a single cluster, or had time series lengths that vary. The latter prohibited a fair comparison of SOMTimeS to K-means. The remaining 112 datasets were used to evaluate the accuracy, execution time, and scalability of SOMTimeS. We fixed the DTW window constraint at 5% of the length of the observation data following earlier recommendations by Paparrizos and Gravano (2016, 2017).

4.4.1 Algorithm Assessment

Accuracy is reported using six assessment metrics, which include the Adjusted Rand Index (ARI) (Santos and Embrechts, 2009), Adjusted Mutual Information (AMI) (Romano et al., 2016), the Rand Index (RI) (Hubert and Arabie, 1985),

Homogeneity (Rosenberg and Hirschberg, 2007), Completeness (Rosenberg and Hirschberg, 2007), and Fowlkes Mallows index (FMS) (Fowlkes and Mallows, 1983). Scalability and execution time of DTW-based clustering algorithms are inversely affected by the length and total number of times series being clustered or classified. As a result, we report the number of DTW computations and execution time as a function of *problem size*, defined as $\sum_{i=1}^{n} |Q|_i$, where $|Q|$ is the length of times series Q, and n is the total number of time series in the dataset. The presence of a few large datasets in the archive makes it more informative to visualize problem size as the natural logarithm (see Figure S1 in Supplementary Material). Finally, for comparison purposes, the same assessment metrics are reported for two of the more popular and robust clustering algorithms that use DTW as a distance measure — 1) K-means and 2) TADPole.

Clustering quality

The performance of SOMTimeS may be quantified in two important ways — 1) comparison to available ground truth observations (i.e., in our case, the class labels accompanying each dataset in the UCR archive), and 2) comparison to other established DTW-based clustering methods.

Table 4.1 summarizes the execution time and six assessment indices associated with the performance of the SOMTimeS, and the two top-performing DTW-based

[1]Adjusted Rand Index (Santos and Embrechts, 2009)
[2]Adjusted Mutual Information (Romano et al., 2016)
[3]Rand Index (Hubert and Arabie, 1985)
[4]Homogeneity (Rosenberg and Hirschberg, 2007)
[5]Completeness (Rosenberg and Hirschberg, 2007)
[6]Fowlkes Mallows index (Fowlkes and Mallows, 1983)

Table 4.1: Comparison of execution time and assessment metrics for SOMTimeS, K-means and TADPole clustering methods using six assessment indices averaged over the 112 datasets in the UCR archive. Note: Assessment indices (usually expressed as values between 0 and 1) have been multiplied by 100; metric averages closer to 100 represent better performance.

Algorithm	**Hours**	**ARI** [1]		**AMI** [2]		**RI** [3]		**H**[4]		**C**[5]		**FMS**[6]	
		avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
SOMTimeS 100-epochs	**98**	**24**	**23**	**30**	**26**	**71**	**16**	**31**	**25**	**35**	**28**	**50**	**19**
K-means - DTW	158	24	24	29	25	71	16	31	27	34	28	51	19
TADPole	1011	16	25	24	27	62	18	25	26	36	31	51	20

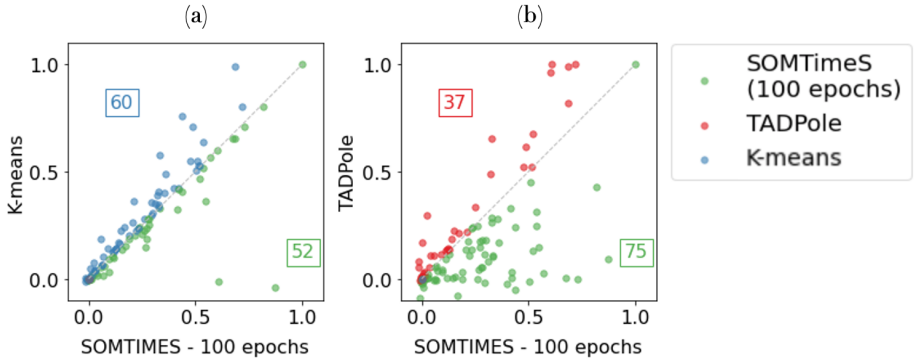


Figure 4.6: ARI scores for SOMTimeS (shown in green) vs. (a) TADPole (red), and (b) K-means (blue) across all 112 of the UCR datasets.

clustering algorithms — K-means and TADPole clustering algorithms. While on average, SOMTimeS has higher assessment indices and lower standard deviations compared to K-means and TADPole, the differences between SOMTimeS and K-means are negligible. Because ARI is recommended as one of the more robust measures for assessing accuracy across datasets (Milligan and Cooper, 1986; Javed et al., 2020b), we plot the ARI scores for SOMTimeS vs. TADPole, and SOMTimeS vs. K-means (Figures 4.6 (a) and (b), respectively) for each of the 112 URC datasets. The green points (75 of the 112 datasets) lying below the 45-degree line of panel (a) represent higher accuracy for SOMTimeS, while ARI scores above the diagonal (shown in red) indicate that TADPole outperforms SOMTimeS for 37 of the 112

datasets. The comparison of ARI scores for SOMTimeS and K-means (Figure 4.6b) shows higher accuracy for SOMTimeS for 52 of the 112 datasets, and lower accuracy for the remaining 60 datasets.

Execution time and scalability

While the assessment metrics are very similar across all three algorithms, SOMTimeS is much faster. When SOMTimeS is implemented on a single CPU, the algorithm takes 98 hours (4 days) to cluster all 112 of the archived datasets. The closest competitor from an accuracy standpoint took 158 hours (6.5 days) when the number of iterations was capped at 10 or until it was run to convergence, whichever was shorter. TADPole, on the other hand, took more than 40 days. All algorithms were executed on same computational machine — dual 20-Core Intel Xeon E5-2698 v4 2.2 GHz machine with 512 GB 2,133 MHz DDR4 RDIMM. Moreover, SOMTimeS is amenable to parallelization and, when SOMTimeS is executed in parallel, the execution time reduces by a factor of the number of CPUs. In this benchmark study, it took 4.9 hours using 20 CPUs (and took only 40 minutes with comparable accuracy (ARI of 0.21 ± 0.23) when the number of SOM epochs was reduced from 100 to 10).

SOMTimeS' scalability with the problem size (i.e., the number and lengths of time series data) is a result of the pruning strategy. We study the effects of the pruning in four ways — 1) percentage of DTW computations pruned as function of time series length, 2) the total number of DTW computations pruned, 3) the scalability as a function of DTW computations performed, and 4) the change in DTW pruning rate over epochs.

Percentage of pruning with respect to the length of individual time series: Because DTW scales with the length of individual time series, we examined the number of DTW computations pruned as a function of time series length. Figure 4.7a shows the percentage of DTW computations pruned for increasing time series length, in linear scale axis for a subset (n=36) of the UCR archived datasets. Here the subset comprises all datasets where the total number of time series is greater than 100 and the length of time series is greater than 500. Figure 4.7b shows the corresponding log-log plot, where the slope approximates the relationship between pruning rate and time series length. This increase in pruning rate is close to the DTW complexity of $((m^2/\log\log m))$, where m is the length of time series (see Figure 4.7c).

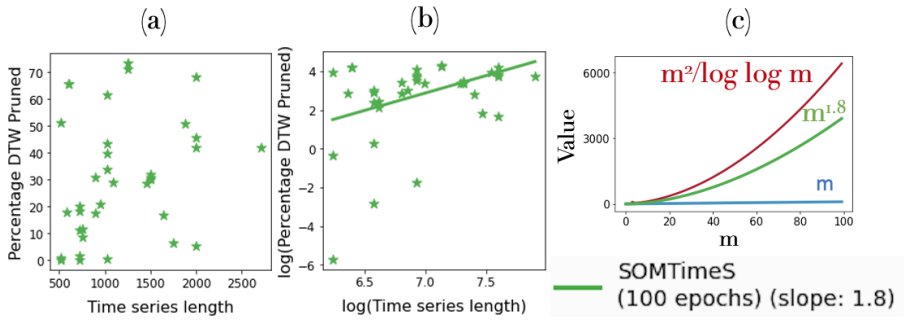


Figure 4.7: Percentage of DTW computations pruned with respect to the time series length shown in (a) linear scale axis, (b) logarithm scale; each green star represents one of the 36 UCR archived datasets, and (c) empirical approximation of the pruning rate as a function of time series length (m).

The total number of DTW computations pruned: Since TADPole is the only algorithm designed to prune unnecessary DTW calculations to speed up clustering, we compared the pruning effects of SOMTimeS with that of TADPole. SOMTimeS (with epochs set to 10 and 100, respectively) pruned more than 50%

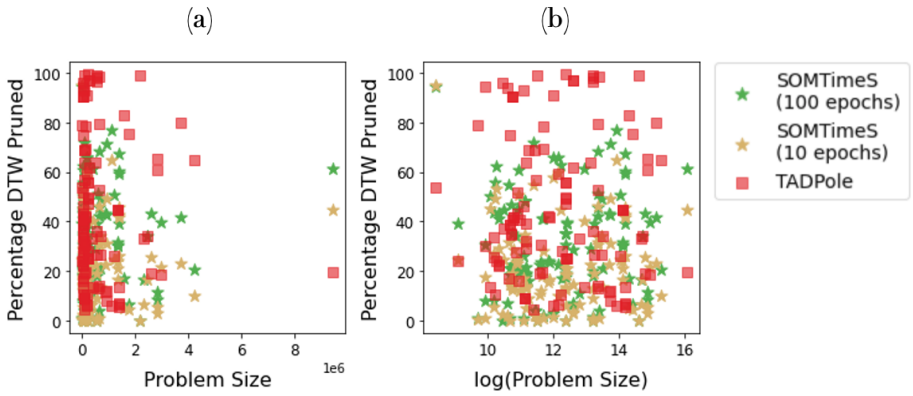


Figure 4.8: The pruning effect of SOMTimeS (10 epochs shown in blown stars), SOMTimeS (100 epochs in green stars) and TADPole (red squares) measured as the percentage of DTW calls pruned during the clustering of a dataset for varying problem size in (a) linear scale axis and (b) natural log axis.

of the DTW calculations for 8 and 21 of the 112 UCR datasets, respectively (see Figure 4.8) whereas TADPole pruned more than 50% of the DTW calculations for 40 of the datasets. Despite this apparent pruning advantage of TADPole, however, its quadratic O(n^2) DTW calculations (as opposed to O(n) in SOMTimeS) results in more DTW computations, particularly for larger datasets.

DTW computations performed: Since TADPole performs O(n^2) DTW calculations, the number of calls to DTW increases quadratically with the number n of input time series. The cutoff (in terms of the number of input time series, n) at which the number of calls to the DTW function in SOMTimeS is less than that of TADPole, depending on the number of epochs. This cutoff is empirically observed to be close to $n = 100$ and $n = 2500$ for 10 and 100 epochs, respectively (see Figure 4.9).

Overall, when clustering over all the datasets in the UCR archive, SOMTimeS computed the DTW measure 13 million and 100 million times (at 10 and 100 epochs, respectively); while TADPole by comparison computed DTW 200 million times (see

Figure 4.9). At a dataset level, SOMTimeS had fewer calls for 88 of the datasets (when using 10 epochs), and 26 of the datasets (for 100 epochs); yet regardless of the epoch size, SOMTimeS achieved higher ARI scores than TADPole.

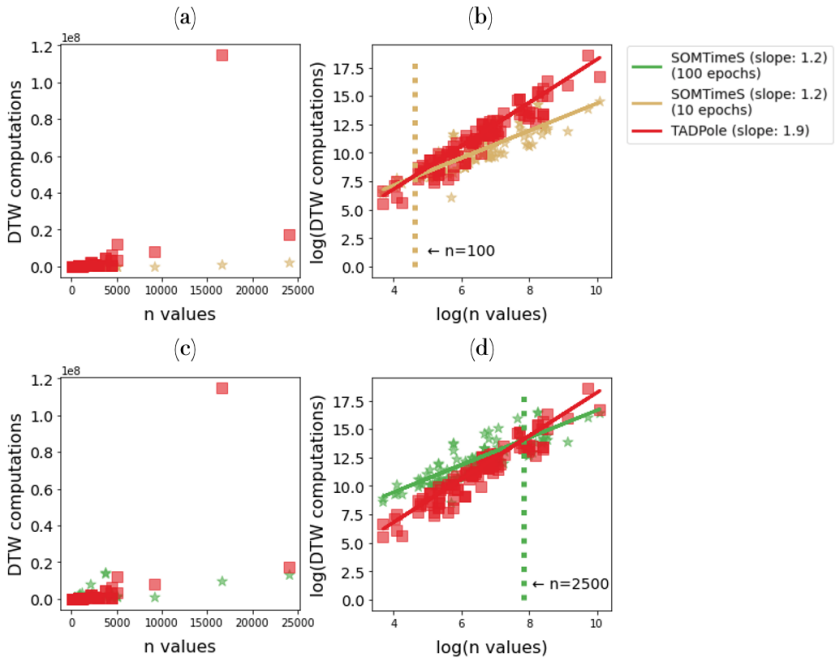


Figure 4.9: Comparison of the number of DTW computations performed for datasets of varying sizes between TADPole (200 million computations total) and SOMTimeS (13 million computations total at 10 epochs, and 100 million computations at 100 epochs) shown on linear scale axis (panels a and c) and corresponding natural-log axis (panels b and d).

Change in the pruning rate over epochs: When we examine the pruning effect as a function of epochs, both the number of DTW calls and the execution time decrease as the number of epochs increases. Figure 4.10a shows the total number of calls to the DTW function made for each dataset, normalized over all epochs. The dashed line represents the average number of calls over all datasets and the shaded region shows the 95% confidence interval. Figure 4.10b shows the corresponding normalized execution time. Both DTW calls and execution time per epoch steadily decrease with increasing number of epochs and iterative updating of

SOM weights. The elbow point, where further epochs resultin diminishing reductions of DTW calculations, is at the 6th epoch. This point is called the swapover point and occurs when the self-organizing map moves from gross reorganization of the SOM weights to fine-tuning of the weights.

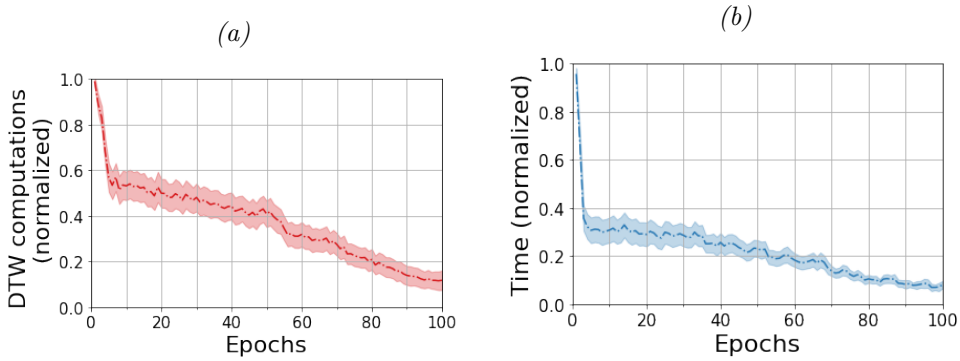


Figure 4.10: Change in the pruning effect of SOMTimeS measured as (a) the number of calls to the DTW functionand (b) the execution time as the number of epochs increases. The dashed line represents the mean value for all datasets after individually normalizing run for each dataset over all epochs. The shaded region corresponds to 95% confidence interval around the mean.

Finally, Figure 4.11 shows how SOMTimeS execution time scales with the problem size. It increases at a lower rate than both TADPole and K-means. TADPole increases at the highest rate, consistent with its O(n^2) complexity of DTW calculations, followed by K-means with a complexity of O($n \times k \times$ number of iterations), where k is the number of clusters. SOMTimeS has complexity of O($n \times k \times e$), where e is the number of epochs. While in theory K-means has complexity of DTW calculations similar to that of SOMTimeS, it does not prune the DTW calculations, and as a result, it is both slower and less scalable. For an empirical point of view, SOMTimeS scales better than existing DTW-based clustering algorithms.

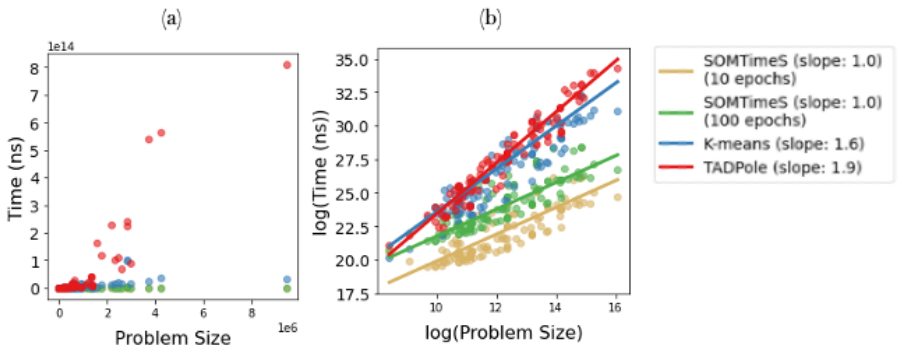


Figure 4.11: Execution time of K-means, TADPole, and SOMTimeS for the select 112 UCR archive datasets in (a) linear scale axis, and (b) natural log axis.

4.5 Application to Serious Illness Conversations

We demonstrate the utility of SOMtimeS in visualizing clustering output by applying it to healthcare communication using actual lexical data collected in the Palliative Care Communication Research Initiative (PCCRI) cohort study (Gramling et al., 2015). The PCCRI is a multisite, epidemiological study that includes verbatim transcriptions of audio-recorded palliative care consultations involving 231 hospitalized people with advanced cancer, their families and 54 palliative care clinicians.

4.5.1 Need for scalability in health care communication science

Understanding and improving healthcare communication requires a method that can measure what actually happens when patients, families, and clinicians interact in large enough samples to represent diverse cultural, dialectical, decisional and clinical contexts (Tulsky et al., 2017). Some features of inter-personal communication, such as tone or lexicon, will require frequent sampling over the course of conversation in order to reveal overarching patterns indicating types of interactions. Discovering patterns (i.e., clusters) of conversations with frequent sampling of features over conversation presents a need for scalable unsupervised machine learning methods. SOMTimeS is equipped to meet the need.

Our previous work suggests that conversational narrative analysis offers a clinically meaningful framework for understanding serious illness conversations (Ross et al., 2020; Gramling et al., 2021), and others have demonstrated that unsupervised machine learning can identify "types of stories" using time-series analysis of lexicon (Reagan et al., 2016). One core feature of conversational narrative, called *temporal reference*, characterizes how participants organize their conversations about things that happened in the past, are happening now, or may happen in the future (Romaine, 1983). This motivates a study of how SOMtimeS can be useful to explore potential clusters of "emporal reference story arcs". Natural language processing methods can reasonably estimate the shape or "arc" of temporal reference

by categorizing verb tenses spoken during a conversation and describing the relative frequency of past/present/future referents over sequential deciles of total words spoken in the conversation (i.e., narrative time). In order to avoid sparse decile-level data in shorter conversations, we selected the 171 of 231 PCCRI clinical conversations as the basis for examining potential clustering.

4.5.2 Data pre-processing: Verb tense as a time series

We used a temporal reference tagger (Ross et al., 2020) to assign temporal reference (past, present, or future) to verbs and verb modifiers in the verbatim transcripts. Specifically, the Natural Language Toolkit (NLTK; www.nltk.org) was used to classify each word in the transcripts into a part of speech (POS), and for any word classified as a verb, the preceding context is used to assign that verb (and any modifiers) to a given temporal reference. Then, each conversation was stratified into deciles of "narrative time" based on the total word count for each conversation, and a temporal reference (i.e., verb tense) time series was generated for each conversation as the proportion of all future tense verbs relative to the total number of past and future tense verbs. The vertical axis in Figure 4.12 represents the proportion of future vs. past talk (per decile), where any value above the threshold (dashed line = 0.5) represents more future talk. Each of the 171 generated time series (see Figure 4.12a) were then smoothed using a 2nd-order, 9-step Savitzky-Golay filter (Savitzky and Golay, 1964) (see Figure 4.12b). Savitzky-Golay filter works by fitting a polynomial over a moving

window (2nd-order polynomial, over a 9-step window in this work) and replaces the data points with corresponding values of the fitted polynomial. Smoothing reduces noise that may result from simplifying assumptions used in modeling the temporal reference time series (i.e., conversational story arcs). We then used SOMTimeS to cluster the resulting conversational story arcs.

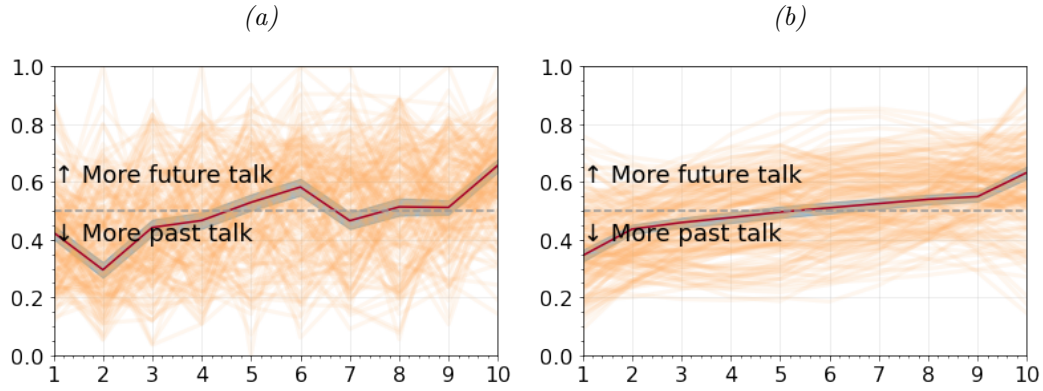


Figure 4.12: Temporal plot showing the (a) raw time series, and (b) smoothed time series for all conversations superimposed in brown; the red line represents the mean values, and the shaded region around the red line represents 95% confidence interval.

4.5.3 Clustering verb tense time series

In applying SOMTimeS to the conversational PCCRI data, we identified $k = 2$ clusters with distinct temporal shapes (see Figure 4.13). Both of the conversational arcs share a temporal narrative with more references to the past at the beginning of the conversation, and more references to the future as the conversation progresses. The proportions of future talk and past talk are more similar at deciles 1 and 10 than at deciles 2 to 9. These conversational arcs are differentiated by the rate at which the narrative changes. Cluster 1 does not enter the "more future talk" region until decile 9, while cluster 2 does much earlier (decile 2). It was expected that the first and last

deciles of the conversations would be more similar given the nature of introductionat the start and farewellat the end of a conversation.

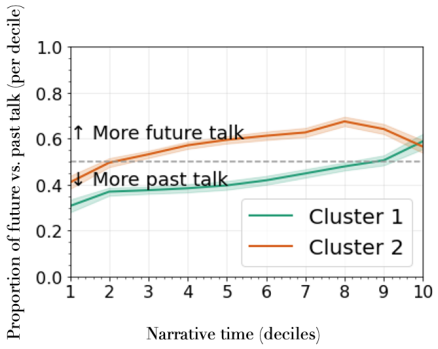


Figure 4.13: Mean values of the proportion of future and past (i.e., verb tense) talks over the narrative time decile for cluster 1 (green) and cluster 2 (brown) with the shaded region representing 95% confidence interval.

Now, let us illustrate how SOMTimeS identifies the number of clusters and visualizes the features that drive those clusters. When the U-matrix is superimposed on the 2-D SOM mesh (see Figure 4.14a), the observations appear to cluster into 2–3 groups based on visual inspection. Keeping the case study objectives in mind, and noting that the 2-D mesh is torodial, we color-coded the $k = 2$ clusters on Figure 4.14b using spectral clustering. In Figure see 4.14c, we superimpose and interpolate the sum of the proportion of future vs. past talk over all deciles in the time series (i.e., conversational arcs of Figure 4.13) in the same 2-D space as the clustered times series.

4.6 Discussion

We present SOMTimeS as a clustering algorithm for time series that exploits the competitive learning of the Kohonen Self-Organizing Map, and the distance bounds of DTW to improve execution time. SOMTimeS contrasts with other DTW-based

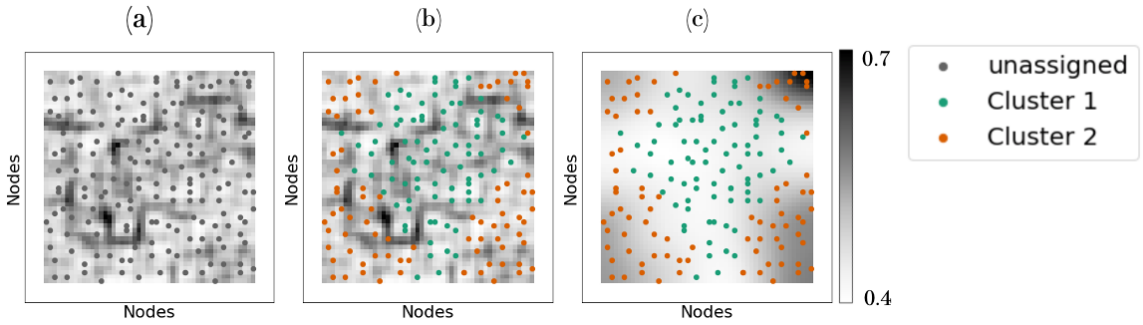


Figure 4.14: Temporal reference time series data from 171 serious illness conversations self-organized on a 2-D map(a) with U-matrix, (b) using spectral clustering, and (c) interpolated sum of the temporal reference time series superimposed on the clustered map.

clustering algorithms given its ability to both reduce the dimensionality of, and visualize input features associated with clustering temporal data. In terms of accuracy, SOMTimeS has similar performance to K-means, which is not unexpected given that both methods use single cluster centroid.

The benchmark experiments in this work are intended to put SOMTimeS in context with state-of-the-art clustering algorithms. Keeping the study objectives in mind, execution times are used to demonstrate scalability, and highlight the feasibility of analyzing large time series datasets using SOMTimeS. K-means is perhaps the most popular clustering algorithm and has been proven time and again to outperform state-of-the-art algorithms; however, because of its simplicity, it lacks the interpretability and visualization capabilities of SOMTimeS. TADPole on the other hand, is a state-of-the-art clustering algorithm that organizes data differently from SOMTimeS (and by extension K-means), as evident from the difference in ARI scores (see Figure 4.6a), and choice of centroids (i.e., density peaks; see Supplementary Material Section 4.8.1). For these reasons, the algorithms tested are not direct competitors of each another and each has advantages in their own right.

SOMTimeS learns (i.e., self-organizes) in an iterative manner such that as the number of SOM epochs increase, the execution time per epoch decreases (see Figure 4.10b), making higher number of epochs feasible. This reduction in time is also directly proportional to the number of calls to DTW function at each epoch. The elbow point (at 6 for SOMTimeS with 100 epochs) indicates quick gains in pruning DTW calculations. This same gain is observed when the total number of epochs is set to 10 or 50 (see Supplementary Material Figure S2). SOMTimeS took 40 minutes to cluster the entire UCR archive using 10 epochs, and less than 300 minutes when the number of epochs was increased 10-fold. Similarly, the largest dataset in terms of problem size took 5 minutes to cluster using 10 epochs, and 35 minutes to cluster at 100 epochs. SOMTimeS demonstrates sub-linear scalability when it comes to increasing the number of epochs. The scalability, fast execution times, and the ease of saving the state (weights) of a SOM make SOMTimeS a potential candidate for an *anytime* algorithm. It possesses the five most desirable properties of anytime algorithms (Zilberstein and Russell, 1995; Zhu et al., 2012).

4.7 Conclusion and Future Work

The explosion in volume of time series data has resulted in the availability of large unlabeled time datasets. In this work, we introduce **s**elf-**o**rganizing **m**aps for **time s**eries (**SOMTimeS**). SOMTimeS is a self-organizing map for clustering and classifying time series data that uses DTW as a distance measure of similarity between time series. To reduce run time and improve scalability, SOMTimeS prunes DTW

calculations by using distance bounding during the SOM training phase. This pruning results in a computationally efficient and fast time series clustering algorithm that is linearly scalable with respect to increasing number of observations. SOMTimeS clustered 112 datasets from the UCR time series classification archive in under 5 hours with state-of-art accuracy. In comparison, other DTW-based algorithms can take anywhere from days to months on the same computing platform. We applied SOMTimeS to 171 conversations from the PCCRI dataset. The resulting clusters showed two fundamental shapes of conversational stories.

To further improve computational efficiency and clustering accuracy, newer and state-of-the-art variations of SOMs may be used that leverage the same pruning strategy in this work. Improving computational time of DTW-based algorithms is an active area of research, and any improvement in computational speed of DTW can be incorporated in SOMTimeS for the unpruned DTW computations. Finally, SOMTimeS is a uni-variate time series clustering algorithm. To create a multivariate time series clustering algorithm, the pruning strategy will have to be revisited to accommodate the variations of DTW for multi-variate time series. SOMTimeS is a fast and linearly scalable algorithm that recasts DTW as a computationally efficient distance measure for time series data clustering.

Acknowledgements

This project was supported by the Richard Barrett Foundation and Gund Institute for Environment through a Gund Barrett Fellowship. Additional support was provided

by the Vermont EPSCoR BREE Project (NSF Award OIA-1556770). We thank Dr. Patrick J. Clemins of Vermont EPSCoR, for providing support in using the EPSCoR Pascal high-performance computing server for the project. Computations were performed, in part, on the Vermont Advanced Computing Core. Data used to illustrate concepts in this paper arise from the Palliative Care Conversation Research Initiative (PCCRI). The PCCRI was funded by a Research Scholar Grant from the American Cancer Society (RSG PCSM124655; PI: Robert Gramling).

Bibliography

Al-Naymat, G., Chawla, S., and Taheri, J. (2009). Sparsedtw: A novel approach to speed up dynamic time warping. In *Proceedings of the Eighth Australasian Data Mining Conference - Volume 101*, AusDM '09, pages 117–127, AUS. Australian Computer Society, Inc.

Alvarez-Guerra, M., GonzĂĄlez-PiĂąuela, C., AndrĂŠs, A., GalĂĄn, B., and Viguri, J. R. (2008). Assessment of self-organizing map artificial neural networks for the classification of sediment quality. *Environment International*, 34(6):782–790.

Begum, N., Ulanova, L., Wang, J., and 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*, pages 49–58.

Bende-Michl, U., Verburg, K., and 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.

Bentley, F., Luvogt, C., Silverman, M., Wirasinghe, R., White, B., and Lottridge, D. (2018). Understanding the long-term use of smart speaker assistants. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(3).

Chu, E., Dunn, J., Roy, D., and Sands, G. (2017). Ai in storytelling: Machines as cocreators.

CRS (2020). The internet of things (iot): An overview. https://fas.org/sgp/crs/misc/IF11239.pdf.

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., and Hexagon-ML (2018). The UCR time series classification archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

De Bie, T., Lijffijt, J., Mesnage, C., and Santos-RodrĂguez, R. (2016). Detecting trends in twitter time series. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, 1(2):1542–1552.

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

Eshghi, A., Haughton, D., Legrand, P., Skaletsky, M., and Woolford, S. (2011). Identifying groups: A comparison of methodologies. *Journal of data science*, 9:271–291.

Evans, D. (2011). The internet of things. how the next evolution of the internet is changing everything. Technical Report MSU-CSE-06-2, Cisco Systems.

Ewen, J. (2011). Hydrograph matching method for measuring model performance. *Journal of Hydrology*, 408(1):178 – 187.

Flanagan, K., Fallon, E., Connolly, P., and 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*, pages 116–121.

Fowlkes, E. B. and Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569.

Gharehbaghi, A. and Linden, M. (2018). A deep machine learning method for classifying cyclic time series of biological signals using time-growing neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4102–4115.

Gold, O. and Sharir, M. (2018). Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Trans. Algorithms*, 14(4).

Gramling, R., Gajary-Coots, E., Stanek, S., Dougoud, N., Pyke, H., Thomas, M., Cimino, J., Sanders, M., Chang Alexander, S., Epstein, R., Fiscella, K., Gramling, D., Ladwig, S., Anderson, W., Pantilat, S., and Norton, S. (2015). Design of, and enrollment in, the palliative care communication research initiative: A direct-observation cohort study. *BMC palliative care*, 14:40.

Gramling, R., Javed, A., Durieux, B. N., Clarfeld, L. A., Matt, J. E., Rizzo, D. M., Wong, A., Braddish, T., Gramling, C. J., Wills, J., Arnoldy, F., Straton, J., Cheney, N., Eppstein, M. J., and Gramling, D. (2021). Conversational stories & self organizing maps: Innovations for the scalable study of uncertainty in healthcare communication. *Patient Education and Counseling*.

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

Hamami, F. and Dahlan, I. A. (2020). Univariate time series data forecasting of air pollution using lstm neural network. In *2020 International Conference on Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, pages 1–5.

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

Iorio, C., Frasso, G., D'Ambrosio, A., and Siciliano, R. (2018). A P-spline based clustering approach for portfolio selection. *Expert Systems with Applications*, 95:88 – 103.

Javed, A., Hamshaw, S. D., Lee, B. S., and Rizzo, D. M. (2020a). Multivariate event time series analysis using hydrological and suspended sediment data. *Journal of Hydrology*, page 125802.

Javed, A. and Lee, B. S. (2016). Sense-level semantic clustering of hashtags in social media. In *Proceedings of the 3rd Annual International Symposium on Information Management and Big Data*.

Javed, A. and Lee, B. S. (2017). Sense-level semantic clustering of hashtags. In Lossio-Ventura, J. A. and Alatrista-Salas, H., editors, *Information Management and Big Data*, pages 1–16, Cham. Springer International Publishing.

Javed, A. and Lee, B. S. (2018). Hybrid semantic clustering of hashtags. *Online Social Networks and Media*, 5:23 – 36.

Javed, A., Lee, B. S., and Rizzo, D. M. (2020b). A benchmark study on time series clustering. *Machine Learning with Applications*, 1:100001.

Johnpaul, C., Prasad, M. V., Nickolas, S., and Gangadharan, G. (2020). Trendlets: A novel probabilistic representational structures for clustering the time series data. *Expert Systems with Applications*, 145:113119.

Keogh, E. (2002). Exact indexing of dynamic time warping. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pages 406–417.

VLDB Endowment.

Keogh, E. J. and 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.

Kohonen, T. (2013). Essentials of the self-organizing map. *Neural Networks*, 37:52–65. Twenty-fifth Anniversay Commemorative Issue.

Kohonen, T., Schroeder, M. R., and Huang, T. S. (2001). *Self-Organizing Maps*. Springer-Verlag, Berlin, Heidelberg, 3rd edition.

Lasfer, A., El-Baz, H., and Zualkernan, I. (2013). Neural network design parameters for forecasting financial time series. In *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, pages 1–4.

Lawrence, R. D., Almasi, G. S., and Rushmeier, H. E. (1999). A scalable parallel algorithm for self-organizing maps with applicationsto sparse data mining problems. *Data Min. Knowl. Discov.*, 3(2):171–195.

Li, K., Sward, K., Deng, H., Morrison, J., Habre, R., Franklin, M., Chiang, Y.-Y., Ambite, J., Wilson, J. P., and P.Eckel, S. (2020). Using dynamic time warping self-organizing maps to characterize diurnal patterns in environmental exposures. *Research Square*.

Li Wei, Keogh, E., Van Herle, H., and Mafra-Neto, A. (2005). Atomic wedgie: efficient query filtering for streaming time series. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, page 8.

Lou, Y., Ao, H., and Dong, Y. (2015). Improvement of dynamic time warping (dtw) algorithm. In *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pages 384–387.

Mangiameli, P., Chen, S. K., and West, D. (1996). A comparison of som neural network and hierarchical clustering methods. *European Journal of Operational Research*, 93(2):402–417. Neural Networks and Operations Research/Management Science.

Mather, A. L. and 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.

Milligan, G. and Cooper, M. (1986). A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate behavioral research*, 21 4:441–58.

Minaudo, C., Dupas, R., Gascuel-Odoux, C., Fovet, O., Mellander, P.-E., Jordan, P., Shore, M., and Moatar, F. (2017). Nonlinear empirical modeling to estimate

phosphorus exports using continuous records of turbidity and discharge. *Water Resources Research*, 53:7590–7606.

Obermayer, K., Ritter, H., and Schulten, K. (1990). Large-scale simulations of self-organizing neural networks on parallel computers: application to biological modelling. *Parallel Computing*, 14(3):381–404.

Paparrizos, J. and Gravano, L. (2016). K-shape: Efficient and accurate clustering of time series. *SIGMOD Record*, 45(1):69–76.

Paparrizos, J. and Gravano, L. (2017). Fast and accurate time-series clustering. *ACM Transactions on Database Systems*, 42(2):8:1–8:49.

Parshutin, S. and Kuleshova, G. (2008). Time warping techniques in clustering time series.

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

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and 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*, pages 262–f–270.

Ratanamahatana, C., Keogh, E., Bagnall, A. J., and Lonardi, S. (2005). A novel bit level time series representation with implication of similarity search and clustering. In Ho, T. B., Cheung, D., and Liu, H., editors, *Advances in Knowledge Discovery and Data Mining*, pages 771–777, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ratanamahatana, C. A. and Keogh, E. (2004). Everything you know about Dynamic Time Warping is wrong. In *Proceedings of the 3rd Workshop on Mining Temporal and Sequential Data*. Citeseer.

Reagan, A. J., Mitchell, L., Kiley, D., Danforth, C. M., and Dodds, P. S. (2016). The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science*, 5(1).

Rodriguez, A. and Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496.

Romaine, S. (1983). Locating language in time and space: William labov (ed.), quantitative analyses of linguistic structure, volume 1. academic press, new york. 271 pp. *Lingua*, 60(1):87–96.

Romano, S., Vinh, N. X., Bailey, J., and Verspoor, K. (2016). Adjusting for

chance clustering comparison measures. *Journal of Machine Learning Research*, 17(1):4635–4666.

Rosenberg, A. and 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*, pages 410–420.

Ross, L., Danforth, C. M., Eppstein, M. J., Clarfeld, L. A., Durieux, B. N., Gramling, C. J., Hirsch, L., Rizzo, D. M., and Gramling, R. (2020). Story arcs in serious illness: Natural language processing features of palliative care conversations. *Patient Education and Counseling*, 103(4):826 – 832.

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

Salvador, S. and Chan, P. (2007a). Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580.

Salvador, S. and Chan, P. (2007b). Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580.

Santos, J. M. and 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*, pages 175–184.

Savitzky, A. and Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639.

Silva, M. and Henriques, R. (2020). Exploring time-series motifs through dtw-som. In *2020 International Joint Conference on Neural Networks, IJCNN*, Proceedings of the International Joint Conference on Neural Networks, pages 1–8, United States. Institute of Electrical and Electronics Engineers Inc.

Somervuo, P. and Kohonen, T. (1999). Self-organizing maps and learning vector quantization forfeature sequences. *Neural Process. Lett.*, 10(2):151–159.

Souto, M. d., Costa, I., Araujo, D., Ludermir, T., and Schliep, A. (2008). Clustering cancer gene expression data: A comparative study. *BMC Bioinformatics*, 9:497.

Tulsky, J. A., Beach, M. C., Butow, P. N., Hickman, S. E., Mack, J. W., Morrison, R. S., Street, Richard L., J., Sudore, R. L., White, D. B., and Pollak, K. I. (2017). A Research Agenda for Communication Between Health Care Professionals and Patients Living With Serious Illness. *JAMA Internal Medicine*, 177(9):1361–1366.

Ultsch, A. (1993). Self-organizing neural networks for visualisation and classification.

In Opitz, O., Lausen, B., and Klar, R., editors, *Information and Classification*, pages 307–313, Berlin, Heidelberg. Springer Berlin Heidelberg.

WorldEconomicForum (2019). How much data is generated each day? Technical report, World Economic Forum.

Wu, C.-H., Hodges, R. E., and Wang, C.-J. (1991). Parallelizing the self-organizing feature map on multiprocessor systems. *Parallel Computing*, 17(6):821–832.

Wu, R. and Keogh, E. (2020). Fastdtw is approximate and generally slower than the algorithm it approximates. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1.

Xi, X., Keogh, E., Shelton, C., Wei, L., and Ratanamahatana, C. A. (2006). Fast time series classification using numerosity reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 1033–1040, New York, NY, USA. Association for Computing Machinery.

Zhu, Q., Batista, G., Rakthanmanon, T., and Keogh, E. (2012). A novel approximation to dynamic time warping allows anytime clustering of massive time series datasets. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 999–1010.

Zilberstein, S. and Russell, S. (1995). *Approximate Reasoning Using Anytime Algorithms*, pages 43–62. Springer US, Boston, MA.

4.8 Supplementary Material

This supplementary material provides additional information on the following aspects of the study:

1. Section 4.8.1 describes the TADPole (Begum et al., 2015) algorithm.

2. Figure S1 show datasets sorted by increasing problem size on arithmetic and log-log scale.

3. Figure S2 shows change in pruning efficiency of SOMTimeS when the total number of epochs are set to 10.

4.8.1 TADPole

TADPole (Begum et al., 2015) is a density based clustering method that uses Density Peaks (Rodriguez and Laio, 2014) as the clustering algorithm and DTW as the distance measure. The Density Peaks algorithm generates cluster centroids (called "density peaks") that are surrounded by neighboring data points that have lower local density and are relatively farther from data points with a higher local density (Rodriguez and Laio, 2014). 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). In this work, when TADPole is used, k is assumed to be known, and 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 and Laio, 2014). TADPole uses upper bound (Euclidean distance) and lower bound (LB_Keogh) to prune unnecessary DTW calculations in the first phase to speed up the clustering. The algorithm has a complexity of O(n^2) where n is the number of time series observations in the input.

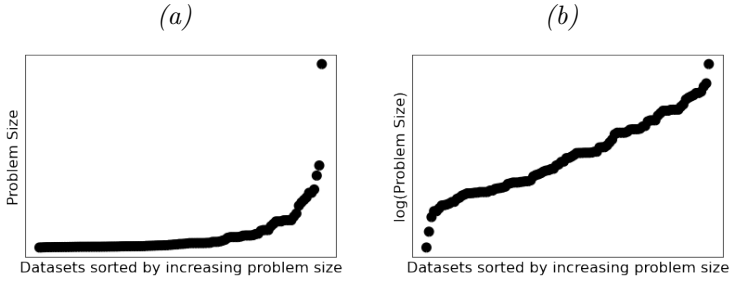
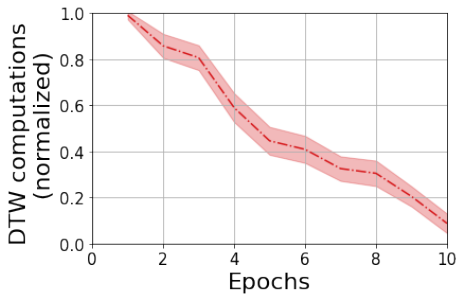


Figure S1: Distribution of all 128 datasets in the UCR archive in terms of (a) problem size and (b) natural log of problem size

.

(a) DTW calculations over epochs



(b) Execution time over epochs

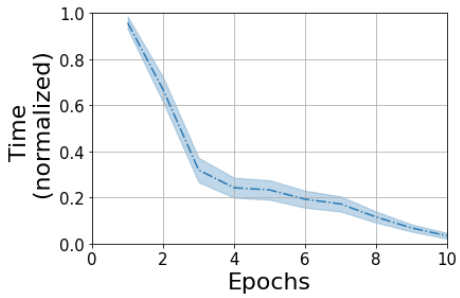


Figure S2: Change in pruning efficiency of SOMTimeS (10 epochs total) as reflected by the calls to DTW function, and execution time over epochs. The dotted line represents the mean value for all datasets after individually normalizing run for each dataset over all epochs. The shaded region corresponds to 95% confidence interval around the mean.

Chapter 5

Conclusion

5.1 Summary

Across the three studies in this dissertation, the common, overarching theme is the clustering of time series with a strong focus on distance measures that enabled time series clustering of real-world observations/events.

In Chapter 2, the time series clustering benchmark study presented used all 128 datasets from the UCR time series classification archive. Observations in the datasets are modeled as time series by default. The data were used in their original form without any preprocessing to keep the benchmark study as unbiased as possible. To ensure the results are useful for a broad range of researchers the benchmark study examined 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). A phased evaluation framework was designed to study the tradeoffs between different algorithms and distance measures. The clustering was evaluated using six popular performance metrics and a proposed measure of spread to assess the variability in performance between two clustering methods. The clustering methods demonstrated high variation

in performance across the datasets. The benchmark study concludes that in the absence of labeled data, the selection of a clustering method requires a thorough understanding of the data, the application at hand and the study objectives.

In Chapter 3, hydrological storm events were modeled as multivariate event time series of river discharge and suspended sediment data. Careful selection of preprocessing routines (i.e., normalization and smoothing) assured that clustering was driven by the shape of the C-Q time series and not by sensor noise or magnitude of the variables. The study used k-medoids as the clustering algorithm and DTW as the distance measure for their respective abilities to overcome outliers and sensor noise. A synthetic hydrological storm event data generator was designed and used to produce time series (event data) for different types of hydrological storm events. **M**ultivariate **e**vent **t**ime **s**eries (METS) clustering was validated using this synthetic storm event data. The METS clustering was then applied to 603 hydrological events (i.e., river discharge and suspended sediment data) acquired through turbidity-based monitoring from six watersheds in the Lake Champlain Basin located in the Northeastern United States; this resulted in identifying four common types of hydrological water quality events. A separate statistical analysis of the events helped identify hydrometeorological features in common with (perhaps drivers) of the event types. METS clustering approach opens up new possibilities for interpreting emergent event behavior in watersheds.

In Chapter 4, feature values extracted as temporal reference from conversations between seriously ill patients and their palliative care team were modeled as story arcs over time. Careful preprocessing assured that features meaningful to serious illness conversations, such as the proportion of future talk relative to past talk, were preserved in the time series. These temporal data were clustered; DTW was selected as the distance measure for its resilience to temporal distortions when aligning the story arcs. To increase the computational efficiency of DTW-based clustering, a new

method, called SOMTimeS (a SOM for TIME Series) was developed and applied to the select conversational feature. SOMTimeS exploits the competitive learning step of SOMs and the distance bounding of DTW. SOMTimeS was tested for accuracy, speed, and scalability on 128 datasets from the same UCR time series classification archive used in the benchmark study of Chapter 2. SOMTimeS accurately clustered all 128 UCR datasets suitable for clustering in under 5 hours, while other competitors took anywhere from 158 to 1011 hours on the same computing platform. SOMTimeS was then applied to the feature value arcs (i.e., past and future verb tense) of serious illness conversations and resulted in identifying two types of conversational stories. Statistical analysis using pre- and post- conversation surveys helped visualize whether the survey data (independent output variables) were correlated with the identified conversation types/shapes with the intent of improving clinician-patient communication. SOMTimeS provides researchers with a powerful algorithm to cluster large time series datasets.

While clustering methods will organize input data into groups, there is no universal standard for optimizing the number of clusters. Clustering methods are used often as a sub-routine for further down-stream tasks (e.g., engine optimization and image processing), which may be used to assess the clustering with respect to the overall arching goals. This in turn may be used to improve the clustering method. As mentioned in Chapter 2, the validity of the clustering results depends on the target research goals as well as the available data.

5.2 Suggested Future Work

There are three primary areas for future work, each corresponding to a chapter of this dissertation. For the clustering benchmark study, presented in Chapter 2, quantifying the optimal number of clusters and precision and recall of clusters remains an active

research problem. External evaluation measures that compare clustering output to ground truth are useful for benchmarking, but not as informative for exploratory analysis. Internal evaluation measures quantify the intra cluster coherence and inter cluster separation by using a distance measure. When an appropriate clustering algorithm and distance measure are selected, both the internal and external measures will quantify the clustering similarity with consistency. In other words, both the internal and external assessment scores will either be low or high. Comprehensive benchmark studies focusing on time series distance measures can help expand the application of novel distance measures and improve our ability to quantify goodness of clusters.

In the METS clustering of Chapter 3, there are three directions for future work. The first relates to the complimentary nature of clustering and classification (e.g., leveraging the advantages of different classification and clustering algorithms to solve real-world problems). We demonstrated METS' strengths in tandem with the readlity accepted hysteresis classification scheme of Williams (1989) by first classifying the hydraulic events based on their hysteresis loop direction, followed by METS clustering. The order of the tandem approach was based on the desire to preserve the timing of the peaks and degree of offset between the two time series (i.e., hydrograph and sedigraph) that are popular in the hydraulic community. Because DTW is not designed to preserve the directional offset in peak timing (e.g., whether the peak of the sedigraph occurs before or after the hydrograph peak) without altering the distance measure, and because the existing hysteresis loop method captures this feature so well, we opted to use the two methods in tandem in order to leverage their respective strengths. While hysteresis loop classification captures the timing between the two peaks, the METS' DTW distance measure captures the vertical distance between the two time series. For instance, the offset of the two time series at the end of a hydrological event provides valuable process-based information on the ability of the

hydrograph to return to baseflow, which relates to the degree of subsurface saturation. In general, however, this leveraging of methodological strengths may be accomplished in other ways. While not applicable given our motivation on sediment transport, if the degree of subsurface saturation was of interest to stakeholder (e.g., because of concerns regarding flooding), then it might be advantageous to reverse the order of METS and Williams' classifications. Alternatively, one could use a hybrid approach (i.e., classify events using both methods in parallel) and weight the classifications based on stakeholder needs.

The second direction for future work using METS involves the methods for preprocessing data, which has a significant impact on the clustered output. The hydrological input data to METS were normalized by the magnitude of the storm event to values between 0 and 1, and standardized in time such that each individual storm event has the same duration. Doing so was an attempt to force the clustering to be driven by the shape of the multivariate time series and not the magnitude or duration of an event. However, a classification scheme that preserves the magnitude and duration of the individual events would have value in stream hydrology applications. As a result, if METS were used without normalizing magnitude and standardizing time in the manner done in this work, the event classification results may very well be different from the classification result presented in this work.

The third direction for future work in METS, involves the choice of the K-medoids algorithm as a partitional clustering algorithm; the latter assigns every event to a certain cluster. In other words, K-medoids will not classify an event as a noisy event (i.e., not assigned to any cluster). Using a modified K-medoids, or replacing it with a density-based clustering algorithm like DBSCAN, will allow for identification of complex type events that are not classifiable and, thereby, generate more meaningful clusters.

The SOMTimeS algorithm presented in Chapter 4 is a computationally efficient

DTW-based clustering algorithm. While already fast and accurate in performance, SOMTimeS can still benefit from advances in learning rate optimization (e.g., momentum) that further reduce the execution time and often help avoid local optima. In addition, different methods for defining the neighborhood of weights or convergence algorithms can be incorporated to tailor SOMTimeS to research studies with different objectives.

Bibliography

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

Aguilera, R. and Melack, J. M. (2018). Concentration-discharge responses to storm events in coastal California watersheds. *Water Resources Research*, 54(1):407–424.

Al-Naymat, G., Chawla, S., and Taheri, J. (2009). Sparsedtw: A novel approach to speed up dynamic time warping. In *Proceedings of the Eighth Australasian Data Mining Conference - Volume 101*, AusDM '09, pages 117–127, AUS. Australian Computer Society, Inc.

Ali, M., Alqahtani, A., Jones, M. W., and Xie, X. (2019). Clustering and classification for time series data in visual analytics: A survey. *IEEE Access*, 7:181314–181338.

Alvarez-Guerra, M., GonzĂĄlez-PiĂąuela, C., AndrĂŠs, A., GalĂĄn, B., and Viguri, J. R. (2008). Assessment of self-organizing map artificial neural networks for the classification of sediment quality. *Environment International*, 34(6):782–790.

Banerjee, A. and Dave, R. N. (2004). Validating clusters using the Hopkins statistic. In *Proceedings of the 2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, volume 1, pages 149–153.

Begum, N., Ulanova, L., Wang, J., and 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*, pages 49–58.

Bellman, R. (1957). *Dynamic Programming*. Dover Publications.

Bende-Michl, U., Verburg, K., and 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.

Bentley, F., Luvogt, C., Silverman, M., Wirasinghe, R., White, B., and Lottridge, D. (2018). Understanding the long-term use of smart speaker assistants. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(3).

Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer.

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

Bholowalia, P. and 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.

Burns, D. A., Pellerin, B. A., Miller, M. P., Capel, P. D., Tesoriero, A. J., and Duncan, J. M. (2019). Monitoring the riverine pulse: Applying high-frequency nitrate data to advance integrative understanding of biogeochemical and hydrological processes. *Wiley Interdisciplinary Reviews: Water*, page e1348.

Burt, T. P., Worrall, F., Howden, N. J. K., and Anderson, M. G. (2015). Shifts in discharge-concentration relationships as a small catchment recover from severe drought. *Hydrological Processes*, 29(4):498–507.

Calinski, T. and JA, H. (1974). A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3:1–27.

Chen, L., Sun, C., Wang, G., Xie, H., and Shen, Z. (2017). Event-based nonpoint source pollution prediction in a scarce data catchment. *Journal of Hydrology*, 552:13–27.

Chu, E., Dunn, J., Roy, D., and Sands, G. (2017). Ai in storytelling: Machines as cocreators.

CRS (2020). The internet of things (iot): An overview. https://fas.org/sgp/crs/misc/IF11239.pdf.

CUAHSI (2019). Consortium of universities for the advancement of hydrologic science, inc. https://www.cuahsi.org.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., and Keogh, E. (2018a). The UCR time series archive. aiXrv 1801.07758.

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., and Hexagon-ML (2018b). The UCR time series classification archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227.

De Bie, T., Lijffijt, J., Mesnage, C., and Santos-RodrĂguez, R. (2016). Detecting trends in twitter time series. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6.

Dierckx, P. (1993). *Curve and Surface Fitting with Splines*. Oxford University Press, Inc.

Ding, H., Trajcevski, G., Scheuermann, P., and Keogh, E. (2010). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26:275–309.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, 1(2):1542–1552.

Du, M., Ding, S., Xue, Y., and 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.

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

Ehret, U. and Zehe, E. (2011). Series distance - An intuitive metric to quantify hydrograph similarity in terms of occurrence, amplitude and timing of hydrological events. *Hydrology and Earth System Sciences*, 15(3):877–896.

Eshghi, A., Haughton, D., Legrand, P., Skaletsky, M., and Woolford, S. (2011). Identifying groups: A comparison of methodologies. *Journal of data science*, 9:271–291.

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

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

Evans, D. (2011). The internet of things. how the next evolution of the internet is changing everything. Technical Report MSU-CSE-06-2, Cisco Systems.

Ewen, J. (2011). Hydrograph matching method for measuring model performance. *Journal of Hydrology*, 408(1):178 – 187.

Faloutsos, C., Ranganathan, M., and Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, pages 419–429.

Flanagan, K., Fallon, E., Connolly, P., and 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*, pages 116–121.

Fowlkes, E. B. and Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569.

Fränti, P. and Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. *Applied Intelligence*, 48(12):4743–4759.

Gellis, A. (2013). Factors influencing storm-generated suspended-sediment concentrations and loads in four basins of contrasting land use, humid-tropical Puerto Rico. *CATENA*, 104:39 – 57.

Gharehbaghi, A. and Linden, M. (2018). A deep machine learning method for classifying cyclic time series of biological signals using time-growing neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4102–4115.

Gold, O. and Sharir, M. (2018). Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Trans. Algorithms*, 14(4).

Gramling, R., Gajary-Coots, E., Stanek, S., Dougoud, N., Pyke, H., Thomas, M., Cimino, J., Sanders, M., Chang Alexander, S., Epstein, R., Fiscella, K., Gramling, D., Ladwig, S., Anderson, W., Pantilat, S., and Norton, S. (2015). Design of, and enrollment in, the palliative care communication research initiative: A direct-observation cohort study. *BMC palliative care*, 14:40.

Gramling, R., Javed, A., Durieux, B. N., Clarfeld, L. A., Matt, J. E., Rizzo, D. M., Wong, A., Braddish, T., Gramling, C. J., Wills, J., Arnoldy, F., Straton, J., Cheney, N., Eppstein, M. J., and Gramling, D. (2021). Conversational stories & self organizing maps: Innovations for the scalable study of uncertainty in healthcare communication. *Patient Education and Counseling*.

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

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

Hamami, F. and Dahlan, I. A. (2020). Univariate time series data forecasting of air pollution using lstm neural network. In *2020 International Conference on Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, pages 1–5.

Hamshaw, S., M. Dewoolkar, M., W. Schroth, A., Wemple, B., and M. Rizzo,

D. (2018). A new machine-learning approach for classifying hysteresis in suspended-sediment discharge relationships using high-frequency monitoring data. *Water Resources Research*, 54:4040–4058.

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

Iorio, C., Frasso, G., D'Ambrosio, A., and Siciliano, R. (2018). A P-spline based clustering approach for portfolio selection. *Expert Systems with Applications*, 95:88 – 103.

Javed, A. (2019a). Dynamic Time Warping. https://github.com/ali-javed/dynamic-time-warping.

Javed, A. (2019b). K-medoids for multivariate time series clustering. https://github.com/ali-javed/Multivariate-Kmedoids.

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

Javed, A., Hamshaw, S. D., Lee, B. S., and Rizzo, D. M. (2020a). Multivariate event time series analysis using hydrological and suspended sediment data. *Journal of Hydrology*, page 125802.

Javed, A. and Lee, B. S. (2016). Sense-level semantic clustering of hashtags in social media. In *Proceedings of the 3rd Annual International Symposium on Information Management and Big Data*.

Javed, A. and Lee, B. S. (2017). Sense-level semantic clustering of hashtags. In Lossio-Ventura, J. A. and Alatrista-Salas, H., editors, *Information Management and Big Data*, pages 1–16, Cham. Springer International Publishing.

Javed, A. and Lee, B. S. (2018). Hybrid semantic clustering of hashtags. *Online Social Networks and Media*, 5:23 – 36.

Javed, A., Lee, B. S., and Rizzo, D. M. (2020b). A benchmark study on time series clustering. *Machine Learning with Applications*, 1:100001.

Jin, X. and Han, J. (2010). *K-Medoids Clustering*, pages 564–565. Springer US, Boston, MA.

Johnpaul, C., Prasad, M. V., Nickolas, S., and Gangadharan, G. (2020). Trendlets: A novel probabilistic representational structures for clustering the time series data. *Expert Systems with Applications*, 145:113119.

Kaufman, L. and Rousseeuw, P. (2008). Origins and extensions of the K-means algorithm in cluster analysis. *Journal Electronique d'Histoire des Probabilites et de*

la Statistique [electronic only], 4:2–18.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley.

Keesstra, S. D., Davis, J., Masselink, R. H., CasalĂ, J., Peeters, E. T. H. M., and Dijksma, R. (2019). Coupling hysteresis analysis with sediment and hydrological connectivity in three agricultural catchments in Navarre, Spain. *Journal of Soils and Sediments*, 19(3):1598–1612.

Keogh, E. (2002). Exact indexing of dynamic time warping. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pages 406–417. VLDB Endowment.

Keogh, E. J. and 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.

Kohonen, T. (2013). Essentials of the self-organizing map. *Neural Networks*, 37:52–65. Twenty-fifth Anniversay Commemorative Issue.

Kohonen, T., Schroeder, M. R., and Huang, T. S. (2001). *Self-Organizing Maps.* Springer-Verlag, Berlin, Heidelberg, 3rd edition.

Lasfer, A., El-Baz, H., and Zualkernan, I. (2013). Neural network design parameters for forecasting financial time series. In *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, pages 1–4.

Latecki, L. J., Megalooikonomou, V., Qiang Wang, Lakaemper, R., Ratanamahatana, C. A., and Keogh, E. (2005). Partial elastic matching of time series. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 4 pp.–.

Latecki, L. J., Megalooikonomou, V., Wang, Q., Lakaemper, R., Ratanamahatana, C. A., and Keogh, E. (2005). Elastic partial matching of time series. In *Knowledge Discovery in Databases*, pages 577–584.

Lawrence, R. D., Almasi, G. S., and Rushmeier, H. E. (1999). A scalable parallel algorithm for self-organizing maps with applicationsto sparse data mining problems. *Data Min. Knowl. Discov.*, 3(2):171–195.

Li, K., Sward, K., Deng, H., Morrison, J., Habre, R., Franklin, M., Chiang, Y.-Y., Ambite, J., Wilson, J. P., and P.Eckel, S. (2020). Using dynamic time warping self-organizing maps to characterize diurnal patterns in environmental exposures. *Research Square*.

Li, Z. and 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, pages 941–944.

Li Wei, Keogh, E., Van Herle, H., and Mafra-Neto, A. (2005). Atomic wedgie: efficient query filtering for streaming time series. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, page 8.

Liao, T. W. (2005). Clustering of time series data: A survey. *Pattern Recognition*, 38(11):1857 – 1874.

Lloyd, C., Freer, J., Johnes, P., and Collins, A. (2016a). Using hysteresis analysis of high-resolution water quality monitoring data, including uncertainty, to infer controls on nutrient and sediment transfer in catchments. *Science of The Total Environment*, 543, Part A:388 – 404.

Lloyd, C. E. M., Freer, J. E., Johnes, P. J., and Collins, A. L. (2016b). Technical Note: Testing an improved index for analysing storm discharge–concentration hysteresis. *Hydrology and Earth System Sciences*, 20(2):625–632.

Lou, Y., Ao, H., and Dong, Y. (2015). Improvement of dynamic time warping (dtw) algorithm. In *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pages 384–387.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297.

Malutta, S., Kobiyama, M., Chaffe, P. L. B., and BonumĂĄ, N. B. (2020). Hysteresis analysis to quantify and qualify the sediment dynamics: state of the art. *Water Science and Technology*. wst2020279.

Mangiameli, P., Chen, S. K., and West, D. (1996). A comparison of som neural network and hierarchical clustering methods. *European Journal of Operational Research*, 93(2):402–417. Neural Networks and Operations Research/Management Science.

Mather, A. L. and Johnson, R. L. (2014). Quantitative characterization of stream turbidity-discharge behavior using event loop shape modeling and power law parameter decorrelation. *Water Resources Research*, 50(10):7766–7779.

Mather, A. L. and 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.

Maulik, U. and Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, 24(12):1650–1654.

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

Milligan, G. and Cooper, M. (1986). A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate behavioral research*, 21 4:441–58.

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

Mohamad, I. and Usman, D. (2013). Standardization and its effects on K-means clustering algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, 6:3299–3303.

Nadal-Romero, E., RegĂźĂŠs, D., and Latron, J. (2008). Relationships among rainfall, runoff, and suspended sediment in a small catchment with badlands. *CATENA*, 74(2):127 – 136.

Obermayer, K., Ritter, H., and Schulten, K. (1990). Large-scale simulations of self-organizing neural networks on parallel computers: application to biological modelling. *Parallel Computing*, 14(3):381–404.

Onderka, M., Krein, A., Wrede, S., Martinez-Carreras, N., and Hoffmann, L. (2012). Dynamics of storm-driven suspended sediments in a headwater catchment described by multivariable modeling. *Journal of Soils and Sediments*, 12(4):620–635.

Paparrizos, J. and Gravano, L. (2016). K-shape: Efficient and accurate clustering of time series. *SIGMOD Record*, 45(1):69–76.

Paparrizos, J. and Gravano, L. (2017). Fast and accurate time-series clustering. *ACM Transactions on Database Systems*, 42(2):8:1–8:49.

Parshutin, S. and Kuleshova, G. (2008). Time warping techniques in clustering time series.

Patil, C. and 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., and Yüceer, C. (2012). Clustering of high throughput gene expression data. *Computers and Operations Research*, 39:3046–3061.

PRISM (2019). PRISM climate group. http://prism.oregonstate.edu. Last

accessed on March 16, 2019.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and 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*, pages 262–f–270.

Ratanamahatana, C., Keogh, E., Bagnall, A. J., and Lonardi, S. (2005). A novel bit level time series representation with implication of similarity search and clustering. In Ho, T. B., Cheung, D., and Liu, H., editors, *Advances in Knowledge Discovery and Data Mining*, pages 771–777, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ratanamahatana, C. A. and Keogh, E. (2004). Everything you know about Dynamic Time Warping is wrong. In *Proceedings of the 3rd Workshop on Mining Temporal and Sequential Data*. Citeseer.

Reagan, A. J., Mitchell, L., Kiley, D., Danforth, C. M., and Dodds, P. S. (2016). The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science*, 5(1).

Roddick, J. F. and Spiliopoulou, M. (2002). A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767.

Rodriguez, A. and Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496.

Romaine, S. (1983). Locating language in time and space: William labov (ed.), quantitative analyses of linguistic structure, volume 1. academic press, new york. 271 pp. *Lingua*, 60(1):87–96.

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

Rose, L. A., Karwan, D. L., and Godsey, S. E. (2018). Concentration-discharge relationships describe solute and sediment mobilization, reaction, and transport at event and longer timescales. *Hydrological Processes*, 32(18):2829–2844.

Rosenberg, A. and 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*, pages 410–420.

Ross, L., Danforth, C. M., Eppstein, M. J., Clarfeld, L. A., Durieux, B. N., Gramling,

C. J., Hirsch, L., Rizzo, D. M., and Gramling, R. (2020). Story arcs in serious illness: Natural language processing features of palliative care conversations. *Patient Education and Counseling*, 103(4):826 – 832.

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.

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

Salvador, S. and Chan, P. (2007a). Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580.

Salvador, S. and Chan, P. (2007b). Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580.

Santos, J. M. and 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*, pages 175–184.

Savitzky, A. and Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639.

Seeger, M., Errea, M.-P., Begueria, S., Arnaez, J., Marti, C., and Garcia-Ruiz, J. (2004). Catchment soil moisture and rainfall characteristics as determinant factors for discharge/suspended sediment hysteretic loops in a small headwater catchment in the spanish pyrenees. *Journal of Hydrology*, 288(3):299–311.

Sherriff, S. C., Rowan, J. S., Fenton, O., Jordan, P., Melland, A. R., Mellander, P.-E., and hUallachĂĄin, D. O. (2016). Storm event suspended sediment-discharge hysteresis and controls in agricultural watersheds: Implications for watershed scale sediment management. *Environmental Science & Technology*, 50(4):1769–1778.

Shokoohi-Yekta, M. and Keogh, E. J. (2015). On the non-trivial generalization of Dynamic Time Warping to the multi-dimensional case. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 289–297.

Silva, M. and Henriques, R. (2020). Exploring time-series motifs through dtw-som. In *2020 International Joint Conference on Neural Networks, IJCNN*, Proceedings of the International Joint Conference on Neural Networks, pages 1–8, United States. Institute of Electrical and Electronics Engineers Inc.

Somervuo, P. and Kohonen, T. (1999). Self-organizing maps and learning vector quantization forfeature sequences. *Neural Process. Lett.*, 10(2):151–159.

Souto, M. d., Costa, I., Araujo, D., Ludermir, T., and Schliep, A. (2008). Clustering cancer gene expression data: A comparative study. *BMC Bioinformatics*, 9:497.

Stryker, J., Wemple, B., and Bomblies, A. (2017). Modeling sediment mobilization using a distributed hydrological model coupled with a bank stability model. *Water Resources Research*, 53(3):2051–2073.

Subbalakshmi, C., Krishna, G. R., Rao, S. K. M., and 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.

Tulsky, J. A., Beach, M. C., Butow, P. N., Hickman, S. E., Mack, J. W., Morrison, R. S., Street, Richard L., J., Sudore, R. L., White, D. B., and Pollak, K. I. (2017). A Research Agenda for Communication Between Health Care Professionals and Patients Living With Serious Illness. *JAMA Internal Medicine*, 177(9):1361–1366.

Ultsch, A. (1993). Self-organizing neural networks for visualisation and classification. In Opitz, O., Lausen, B., and Klar, R., editors, *Information and Classification*, pages 307–313, Berlin, Heidelberg. Springer Berlin Heidelberg.

UNEP (2021). Why low-cost sensors? opportunities and challenges. https://www.unep.org/explore-topics/air/what-we-do/monitoring-air-quality/why-low-cost-sensors-opportunities-and.

Vaughan, M. C. H., Bowden, W. B., Shanley, J. B., Vermilyea, A., Sleeper, R., Gold, A. J., Pradhanang, S. M., Inamdar, S. P., Levia, D. F., Andres, A. S., and et al. (2017). High-frequency dissolved organic carbon and nitrate measurements reveal differences in storm hysteresis and loading in relation to land cover and seasonality: high-resolution doc and nitrate dynamics. *Water Resources Research*, 53:5345–5363.

Wemple, B. C., Clark, G. E., Ross, D. S., and Rizzo, D. M. (2017). Identifying the spatial pattern and importance of hydro-geomorphic drainage impairments on unpaved roads in the northeastern usa. *Earth Surface Processes and Landforms*, 42(11):1652–1665.

Wendi, D., Merz, B., and Marwan, N. (2019). Assessing hydrograph similarity and rare runoff dynamics by cross recurrence plots. *Water Resources Research*, 55(6):4704–4726.

Williams, G. P. (1989). Sediment concentration versus water discharge during single hydrologic events in rivers. *Journal of Hydrology*, 111(1):89–106.

Williams, M. R., Livingston, S. J., Penn, C. J., Smith, D. R., King, K. W., and Huang, C.-h. (2018). Controls of event-based nutrient transport within nested headwater agricultural watersheds of the western Lake Erie basin. *Journal of Hydrology*, 559:749–761.

WorldEconomicForum (2019). How much data is generated each day? Technical report, World Economic Forum.

Wu, C.-H., Hodges, R. E., and Wang, C.-J. (1991). Parallelizing the self-organizing feature map on multiprocessor systems. *Parallel Computing*, 17(6):821–832.

Wu, R. and Keogh, E. (2020). Fastdtw is approximate and generally slower than the algorithm it approximates. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1.

Wu, X. and Kumar, V. (2009). *The Top Ten Algorithms in Data Mining*. Chapman & Hall/CRC, 1st edition.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., and Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37.

Wymore, A. S., Leon, M. C., Shanley, J. B., and McDowell, W. H. (2019). Hysteretic response of solutes and turbidity at the event scale across forested tropical montane watersheds. *Frontiers in Earth Science*, 7:126.

Xi, X., Keogh, E., Shelton, C., Wei, L., and Ratanamahatana, C. A. (2006). Fast time series classification using numerosity reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 1033–1040, New York, NY, USA. Association for Computing Machinery.

Zhu, Q., Batista, G., Rakthanmanon, T., and Keogh, E. (2012). A novel approximation to dynamic time warping allows anytime clustering of massive time series datasets. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 999–1010.

Zilberstein, S. and Russell, S. (1995). *Approximate Reasoning Using Anytime Algorithms*, pages 43–62. Springer US, Boston, MA.

Zuecco, G., Penna, D., Borga, M., and van Meerveld, H. J. (2016). A versatile index to characterize hysteresis between hydrological variables at the runoff event timescale. *Hydrological Processes*, 30(9):1449–1466.