Time Series Clustering: A Survey

Philipp Beer*
* Graduate Program in Data Science (UNIC), Nicosia, Cyprus

***Abstract*—In this literature review we focus on analyzing and reporting on the clustering of time series data. Machine learning touches many diverse fields. But a particular field that captures a lot of activities is the analysis and utilization of time series data. Applications in this area very diverse ranging from forecasting to reporting, analytics as well predictive maintenance. Time series cover many application areas ranging from the natural sciences, to medicine, industry, finance to robotics. All areas have in common that the utilization of time series for a particular task is quite challenging as it requires lots of considerations and in many cases only a limited set of historical data is available to achieve any given machine learning task. In consequence, an important task is to understand the intrinsic properties of any given time series often times in the context of adjacent time series part of a particular domain set. Clustering is an unsupervised learning technique that allows to evaluate the similarity of time series and group them into alike clusters. This paper reviews the literature on this topic and describes the elements required for performing time series clustering as well the results that have been achieved.**

***Index Terms*—clustering, time series, distance metrics, assessment metrics**

I. Introduction

This paper is organized in the following way: First we will discuss what is clustering and what constitutes a time series alongside how an approach to clustering of time series looks like. We will also distinguish between different representations a time series can take when being presented to a clustering algorithm. After these overviews we will review the different distance metrics that are available for use in clustering algorithms. Then we will take a look at the different categories of clustering algorithms used in time series clustering. The final process step will be to highlight the different assessment metrics that are available in evaluating clustering results and how they can be compared. We will finish the paper by reviewing the limitations encountered in the papers analyzed for this review.

A. Clustering

Clustering is an **unsupervised** learning technique that allows to learn from data without an instructors input and reveal useful patterns that can be exploited when using the clustered data. Estivill-Castro [1] argues that the definition of what de clustering is not clear cut and depends on the viewpoint of the analyst or practitioner. According to the author this can range from (1) looking top-down and segmenting a population of individual data points into a number of homogeneous subgroups, (2) looking bottom-up by finding groups through some criterion of similarity, (3) a distinction whether any pair of elements belong into the same cluster or not. In [2] the authors add distinctions how the clustering algorithms can be distinguished further by defining how my clusters a data point can belong to. It can be **exclusive** - belonging to a single cluster only, **overlapping** or **non-exclusive** - meaning it can belong to more 1 or more clusters simultaneously, and **fuzzy** - where it each points is assigned a probability value between 0 and 1 indicating weight of its membership in a cluster. Another additional dimension of clustering algorithms is whether they completely or on partially cluster the data points. If not all data points are associated with a cluster it is partial clustering and the remaining data points are summarized in a "noise" cluster.

B. Time series

In [3] time series are defined as sequence composed by a series of continuous, real-valued elements. They share the same challenges as other high dimensional data in that it quickly requires high computational power to process and suffers from the "curse of dimensionality" [4]. According to [5] the utility of clustering time series in the real world comes from being able to detect **anomalies**, **novelties**, or **discord** as one can discover dynamic changes in time series, generate predictions and recommendations or discover interesting patterns related to the underlying data generating processes.

C. Simlarity measure in time series

Similarity measures in clustering algorithms oftentimes are computed pair-wise in clustering algorithms. This usually requires same length time series in order to be able to compute similarity. As an additional challenge this computation is impacted by the inherent noise and potential outliers in time series [5]. In part this challenge can be addressed by the data portions that are being clustered, specifically (1) whole time-series clustering, (2) subsequence time-series clustering, and (3) time point clustering [5]. The studies examined in this review are mostly focused on whole time series clustering. For whole series clustering an additional distinction is made by Aghabozorgi *et al.* [5]. The available dimensions for the clustering can be seen in table I. Depending on the method of choice a suitable distance measurement is applied to generate the clusters.

D. Components in time series clustering

When considering whole time series clustering Aghabozorgi *et al.* [5] started to develop a mental model for the dimensions to be considered in time series clustering: (1) time series representation - in which way the time series is represented in the clustering algorithm, (2) similarity or distance measure

TABLE I
TIME SERIES PROPERTIES USED FOR CLUSTERING [5]

Clustering properties	Description
shape-based	Raw data shapes of two time-series are matched to best extend possible by stretching or contracting the time axes.
feature-based	Raw time series is converted into a lower dimensional vector of features. It allows for equal length feature vectors.
model-based	Raw time series is transformed into a parametric model for each time-series

- which metric is used to express distance or similarity, (3) clustering prototypes - how series are assigned to a cluster, and (4) clustering itself - execution of the clustering and evaluation of the results. This proposed components cannot be used as framework or high-level algorithm to clustering as the authors express that some or all of the mentioned components are executed. How to make the distinction when a component is required is not clarified. However, it is still useful to consider these criteria when approaching a time series clustering problem.

II. REPRESENTATION OF TIME SERIES IN CLUSTERING ALGORITHMS

A. Overview

As stated in section I time series bring along the challenges associated with high dimensional data which makes their analysis more challenging. In particular requirements for memory and computational power are heightened with many dimensions. Additionally, depending on the distance measure employed, results are highly sensitive to distortions in the data [5]. One proposal is to use dimensionality reduction techniques to extract meaningful features [5], [6]. Javed *et al.* [7] choose a different approach of addressing this challenge by restricting the aspects of the used time series with respect to their analytics goal of comparing clustering techniques. They choose time series that are equal in length and they work towards pre-identified classes of the dataset. This is a luxury that usually is not available in real world use cases. Another challenge here is to consider whether the pre-defined clusters are probably only one valid set of clusters that may be different for different analytical tasks more in line with the definition of Estivill-Castro [1] of what clustering is. Given that the clusters to be found are predefined, for example in a binary classification, the raw time series can be used while the parameters of the chosen clustering algorithm and the distance measure can be modified to optimize the results. In other contexts where there are more unknowns feature extraction may be vital in order to get closer to the desired clustering results. We found no framework or decision guidance in the literature when which representation is appropriate or should be avoided.

B. Time series representation categories

Aghabozorgi *et al.* [5] provide a categorization for the different time series representation methods differentiating between: (1) data-adaptive - a representation model that minimizes an error when reconstructing the series from this model, (2) non-data adaptive - a representation of fixed-size segments, (3) model-based - a representation with stochastic models , and (4) data dictated - transforming obtaining a compressed discrete representation.

C. Time series Feature extraction

One option mentioned in I is the feature-based representation which is a method reducing the dimensionality of a time series and thereby reducing the computational cost of clustering the different time series and - given that meaningful features are chosen - improve the utility of the clustering results. In 2012 an experimental study was conducted to compare different representation methods for reducing the dimensionality of time series [8]. The following methods were identified: (1) Discrete Fourier Transformation (DFT), (2) Discrete Wavelet Transformation (DWT), (3) Discrete Cosine Transformation (DCT), (4) Singular Value Decomposition (SVD), (5) Adaptive Piecewise Constant Approximation (APCA), (6) Piecewise Aggregate Approximation (PAA), (7) Chebyshev polynomials (CHEB), and (8) Symbolic Aggregate approXimatio. One method missing from that study is Principal Component Analysis (PCA) which is based on matrix decomposition and allows to extract the the components relevant for the variance within a series in an ordinal way. This is helpful as the resulting dimensionality can be varied to a desired level for the analysis task at hand. In other words, irrelevant components can be removed from the results [6]. Another noteworthy method surveyed by Ali *et al.* [6] in the section of Multidimensional Scaling (MDS) is the analysis of k-grams. The method is outlined in [9] and utilizes the idea of creating sub-sequences of the time series of length k. In a second step the occurrence of each k-gram is counted for each of the time series and serve as a feature vector. From first glance this approach lends itself to time series with discrete values. In order to utilize it on real valued series one could transform the real values into adequately spaced ranges and utilize this method of feature extraction. This would bear the risk of information loss from the series if the range-thresholds are chosen inadequately.

III. DISTANCE METRICS

A. Overview

Distance metrics are the cornerstone of clustering algorithms. Depending on the utilized representation method of the time series (see table I) different distance methods may be appropriate. When choosing the shape-based representation method stable distance metrics put prerequisites on the raw time series like having equal lengths and being devoid of large outliers and having limited noise. An alternative to this is to utilize approximate metrics. This can either be achieved directly through the used metrics, for example dynamic time warping (DTW) or by changing the time series representation into a feature- or model-based representation. With the modified representation otherwise less useful metrics like Euclidean distances can be applied again. Aghabozorgi *et al.* [5] indicate that the most popular distance metrics are the (1) Hausdorff distance, (2) modified Hausdorff distance, (3) HMM-based distance, (4) dynamic time warping, (5) Euclidean distance, (6) Euclidean distance in a PCA subspace, and (7) longest common sub-sequence. The study [6], [7] utilizes the Euclidean distance and DTW. Additionally Javed *et al.* [7] employ shape-based distance. It measures the cross-correlation between two time series and indicates with a scale of 0.0 to 2.0 whether a series is identical (0.0) or maximally different (2.0). To make the metric scale invariant the underlying time series need a transformation to be z-normalized. In [4] the distances measures chosen are elastics measures DTW and also a shape-based distance. Another interesting study from Agrawal *et al.* that was seeking to implement an efficient similarity search was relying solely on Euclidean distance [10].

B. Limitations

In general the authors acknowledge the challenges that these distance metrics are accompanied by. For example, it is commonly accepted that simple Euclidean distance is very sensitive to unique features in the time series like outliers and noise. Further, for simple Euclidean distance same length time series are required to be able to compute comparable distance metrics. Other metrics address these limitations in part. For example dynamic time warping is able to overcome the different length time series by warping two series before computing the actual distance between them. This in turn brings its own set of challenges, for example that the warping happens around local extremes that are not indicative of the actual shape of the underlying series. In the area of distance metrics many different approaches have delivered useful results but a framework for choosing the appropriate metric for a specifc task has not emerged.

C. Direction of research

As a consequence the research around distance metrics in part aims to identify new metrics or improve upon existing ones by removing some of the addressed remedies. A case in point is the shapeDTW matching algorithm proposed by Zhao and Itti [11] to improve upon the DTW method by reducing the false alignment around local extremes. It achieves this by matching local descriptors - which can be understood as local windows around a data point describing the neighborhood around the point - based on DTW, meaning that also temporal consistencies are enforced [11]. A restriction that has not been overcome with this updated approach is the quadratic time complexity ($\mathcal{O}(N^2)$). This limits its utility especially with large data sets. The performance area of dynamic time warping is also addressed. Mueen and Keogh [12] discuss the variants of DTW and propose various optimization techniques for an efficient usage of DTW [12]. The authors also argue that if the parameter for the warping window or the normalization is setup inadequately the results are not reaching the desired quality and point to research that addresses these topics.

Other papers focus on the development of new distance measures. In [13] the authors develop a specific similarity measure optimized for time-series clustering, based on the slope of each segment of time-series, and Euclidean distance implemented via DTW. It relies on a dimensionality reducing transformation of the time series data (APSOS) proposed in [14].

IV. Clustering algorithms

According to Ali *et al.* [6] the aim of clustering is to define a grouped structure of similar objects in unlabeled data based on their similar features. The features used for the clustering do not provide "any information about the appropriate group for its objects, they only describe each object in the dataset, assisting clustering algorithms to learn and extract useful information for their structure" [6]. The types of data that can be used for clustering is described in table I. The clustering algorithm categories are summarized in table II.

TABLE II
Overview clustering algorithm types

Algorithm category	general description
Partional	aims to find prototype-based clusters and associates the time series with it; number of clusters is predefined (k)
Hierarchical	each time series starts as individual cluster and with iterations is associated with other time series to form a cluster by some measure of closeness
Density-based	associates time series into a cluster based on a density measure
Grid-based	quantizes feature space and segments time series into created intervals

Depending on the focus of the particular studies the time series data is treated differently. The benchmark study done by Javed *et al.* [7] uses the raw time series to verify algorithms from all categories except grid-based algorithms [7]. This means no pre-processing or modifying the time series has been conducted. Euclidean distance is one of the test distance metrics therefore only same length time series can be used. Furthermore, that analysis is restricted in the sense that the clusters to be found are pre-defined. An additional restriction is that the tested datasets need to have minimum two clusters. In essence, the study aims to find those pre-determined labels of the the time series if the time series meets the outlined criteria. This in some sense is at odds with the definition of clustering as they aim to reach a desired target and the success computed against that. Other studies, like the ones outlined in [6] look at algorithms comparatively based on assessment metrics and other criteria like computational complexity and performance of the algorithms.

A. Partitioning methods

Partitioning methods encompass algorithms that group unlabeled data into different groups. The number of groups *k* is determined by the operator of the algorithm and usually is a parameter that needs to be found through an interactive process. The clusters are defined by iteratively found prototypes which the individual time series get associated with. Examples of partitioned algorithms are K-Means and K-Medoids. In [6] the authors distinguish between hard and fuzzy partioning algorithms. The difference is created by either associating a particular series with a singular cluster or associating the time series with all clusters through a probability value. All probabilities for a single series need to sum up to 1. These algorithms are particularly unfit of assigning meaningful cluster with raw time series of unequal length and most of the time require some type of data transformation to do be able to achieve useful results.

1) Hard/crisp clustering methods: A widely adopted hard clustering method is K-Means which requires a partitioning-defined k. k centers are randomly initiated in the data space and subsequently all time series associated with the closest cluster. After this the centroid of each k is re-computed by taking the mean of that cluster and the cycle of association is started anew. This continues until no cluster association changes any longer or a pre-determined amount of cycles has been completed. This algorithm has been adopted by most of the reviewed papers. A variant of this algorithm is the K-Medoid algorithm in which the centroid is exchanged with a medoid, meaning the time series that is representative of the middle. This change achieves that the computed center is actually associated with what can be called a prototypical instance of the cluster. The centroid in K-Means usually is not associated with an existing time series. These algorithms optimize the results by minimizing the intra-cluster distances and maximizing the inter-cluster distances. K-Medoids is comparatively more stable than K-Means as it can be affected less by outliers [6].

There are many variations of these hard clustering methods with various distance measures analyzed by researchers. Aghabozorgi *et al.* [5] reference several in their study. One interesting example is the approach introduced by Abdulla *et al.* [15] and further explored in this study [16]. The authors build a cross-words reference template (CWRT) where first all series are aligned by DTW to a medoid. From this the average time series is computed. The advantageous property of this approach is that it is irrelevant in which order the time series are processed.

2) Fuzzy clustering methods: The main difference between fuzzy and crisp clustering algorithms is that the association to a cluster is not singular but measured by changing degrees of association, meaning that each time series is associated with different - possibly all - clusters with varying degrees of probabilities. The probabilities sum up to 1. A well researched implementation of fuzzy clustering is **fuzzy C-Means**. The variant **fuzzy K-Medoids** is also subject of research. They are both extended versions of their respective counterparts K-Means and K-Medoid [6]. Fuzzy algorithm are more computationally intensive than their hard clustering counterparts.

B. Hierarchical methods

Hierarchical methods can be sub-divided into *bottom-up* and *top-down* approaches. They are based on creating a tree-structure, either by starting out for each time series as individual cluster and creating groups stepwise by bringing the closest time series or existing clusters into a new group based on the chosen metric (bottom-up). This process continues until all series are combined or a termination condition is met. The opposite approach is to start with one big cluster and separate time series out until each series is separated or a termination condition is reached (top-down).

The association with a cluster is also achieved via the distance measure between all the elements or groups already created. Then the closest elements and/or groups are combined to form a new group that unionizes the underlying time series. The way distance between an element and a group or between groups is computed can be sub-divided into single-link, average-link and complete-link clustering [2].

1) Challenges: Ali *et al.* [6] point out that a challenge for hierarchical clustering algorithms is that adjustments are not feasible after a decision (integration or separation) is made. This is in contrast to the partioning algorithms as they can update a cluster membership once a new centroid or medoid is computed. There are several studies that address these challenges [17]–[19]. Another issue with hierarchical clustering is the computational complexity of $\mathcal{O}(N^2)$ pointed out by Keogh and Lin [20]. This prohibits its use with very large datasets.

2) Advantages: The resulting tree can be analyzed and the integration of separation process visualized via a dendogram. In consequence, choosing the appropriate number of clusters for an analytical task can be identified visually [6] and no number of clusters need to be provided prior to execution of the algorithm.

C. Model-based methods

Model based methods are aimed at learning the model from the data by updating the model parameters. Both Aghabozorgi *et al.* [5] and Ali *et al.* [6] review concepts that implement this approach. Self-organizing maps (SOM) appear to be the most popular method. They can be understood as neural networks implemented as **unsupervised** learning method [21] in which the input data is mapped to field of neurons and for each input vector the most similar neuron in the lower dimensional space (often 2d) is identified and the weights of the winner and the neighboring neurons is updated. The number of nodes represent the different clusters and similar input vector tend to map to the same neurons. The SOM algorithm is stable and its applications are versatile [6].

Other well known model-based clustering algorithms for time series are (1) polynomials, (2) Gaussian mixture models, (3) ARIMA, (4) Markov chain, and (5) Hidden Markov models [5].

1) Challenges: For SOM Warren Liao [22] identified that the utilization of time series of unequal length does not work well as it is difficult to define the proper dimension of the weight vectors for the mapping of the input data. Additionally, parameters need to be set by the user which may be inadequate and spoil the clustering results [5]. Lastly, its computational complexity is a hindrance with large datasets.

2) Advantages: Due to the ability to reduce the dimensionality of the feature space the clustering results can be easily visualized.

D. Grid-based methods

Grid-based clustering algorithms are built on the process of quantizing the feature space of the data into non-overlapping hyper-rectangles [23]. These resulting hyper-rectangles are called *cells*. The grid is made up of these cells. Each dimension of the time series feature space is segmented into a range of intervals. Each feature vector from the time series is then mapped to the corresponding interval in each of its dimensions. For each cell the respective metrics like mean, variance and probability distribution is computed based on the data it contains.

1) Challenges: An issue with this method is that no relationship between the different grids is created [23]. This has a negative impact on the quality of the clusters. The interval ranges of the cells to be created need to be determined and is not inferred via the algorithm. An interesting research question is whether there is a method of creating meaningful intervals based on the underlying data.

2) Advantages: The clustering can be executed in a single pass computation $\mathcal{O}(N)$. And also querying the results of this very fast ($\mathcal{O}(k)$), where k represents the number of grids in the feature space. And the statistical data for each cell is available through the prior computation [23]. The algorithm assists and interacts with scatter plotting as visualization technique which is also a visual aid in making clustering decisions [6].

E. Density-based methods

The most highly used method of this category is DBSCAN, a density function that requires two parameters. The *eps*-neighbourhoud surrounding the point and *minPts* required to be declared a cluster. The method is proposed by Ester *et al.* [24]. A cluster is expanded if many points are spaced closely together in the predefined range. If the neighbors of cluster center are dense as well, the cluster is expanded. The expansion of a cluster ends if the points start being separated or the density of neighbors is lower. Other commonly used algorithms in this category are (1) OPTICS and (2) LOF. Another variation discussed in [7] is *Density Peaks* that does not require the *minPts* parameter but instead a definition of how many clusters *k* are to be identified.

1) Challenges: Interestingly, Aghabozorgi *et al.* [5] point out that density-based clustering has not been used broadly for time-series data clustering because of its high complexity. Additionally the user needs to determine the *eps* and *minPts* parameter. This of course is a hindrance if the feature space is not well understood. Clusters with varying densities also pose a challenge to this algorithm [6] as the distance parameter is fixed. Possibly further analysis with various time series will allow to create a better understanding of its utility and disadvantages in time series clustering.

2) Advantages: Advantages that separate this clustering method from other algorithms is that it can deal very well with non-globular shapes [23] and its execution speed. Therefore it can be applied to well to a wider spectrum of data both in shape and in size. Depending on the used algorithm variation there is no need to pre-define the number of clusters but they emerge from the data itself. Choosing the *minPts* parameter also allows to identify noise and outliers in the data [6].

V. Assessment Metrics

The evaluation of clustering algorithms for time series is a complex activity as the clustering algorithms themselves as well as the utilized distance metrics have many moving parts, meaning parameters that require adjusting and fine-tuning. Further Keogh and Kasetty [25] point out in their important survey that there are usually experimental flaws, in particular data bias and implementation bias in the usage of these algorithms. A negative side effect of this is that results created in other papers have very little generalizability to them. And what is worse that the shown results are not reproducible in the real world. As solution the authors propose that: (1) algorithms should be tested on wide range of datasets (trained on a train set and tested on a test set), (2) efforts need to be taken to avoid implementation bias, (3) novel similarity measures need to be compared to simple and known metrics, like Euclidean or DTW to verify any claims, and (4) data and code need to be shared to allow for review in the scientific community.

1) Good results with respect to what: Hiu *et al.* [26] in their paper point out that determining the extracted clusters of algorithms is difficult when labels for the data sets are absent. In such a case it is in the hands of the analyst to determine the results. Usually it is based the targets and the achievement of those in the subsequent data science steps that follow the clustering. This can be subjective with respect to the analysis task [5]. Even when labels are known for a dataset the results are only meaningful with respect to those pre-determined classes and not all the other potential ends to which the clustering results can be applied.

2) Common Metrics: Aghabozorgi *et al.* [5] provide a helpful classification of assessment metrics which can be found in tables III (external metrics) and IV (internal metrics). The different metrics mentioned are widely employed and have their own properties which depending on the task can be advantageous or harmful. Therefore, no ordinal arrangement of their utility exists. Furthermore, no framework could be found that instructs their utility for particular use cases.

3) External Indexes: These metrics are applied when using external indexes of the data set validate the performance [27]. These external indexes are a classification of the dataset that exists outside the clustering algorithm. It does not necessarily

TABLE III
EXTERNAL EVALUATION METRICS ACCORDING TO [5], [7]

Measure
Cluster Purity
Cluster Similarity Measure (CSM)
Rand Index (RI)
Adjusted Rand Index (ARI)
Adjusted Mutual Information (AMI)
Completeness
F-measure
Normalized Mutual Information (NMI)
Entropy
Jaccard index

TABLE IV
INTERNAL EVALUATION METRICS ACCORDING TO [5], [7]

Measure
Sum of Squared Errors (SSE)
Silhouette Score
Davies-Bouldin index
I-index
Calinski-Harabasz
Dunn index
R-squared index
Hubert-Levin (C-index)
Krzanowski-Lai index
Hartigan index
Root-Mean-Square Standard Deviation (RMSSTD) index
Homogeneity index
Semi-Partial R-squared (SPR)
Distance between two cluster (CD) index
Weighted inter-intra index
Separation Index

mean that those external indexes are a real ground truth. It can be also just comparison to other clustering results. They indicate the degree of matching between two partitions, where one exists outside of the clustering algorithms and the second is the one generated via the method. Oftentimes the truth is human created [5].

One example, *cluster purity* associates a found cluster with the external class to which most of its contained elements belong. The number of correctly associated elements is divided by all elements in the cluster [5]. The *Jaccard score* and *F-measure* are also frequent metrics for external indexes. Another very interesting measure is the *Rand index* which measures the amount of agreement between two clusters. In the *Adjusted Rand Index* form it adjusts for the chance grouping of elements as well.

The mentioned metrics are very informative but only of limited value if no ground truth to validate against is available for the clustered data set. The study by Javed *et al.* [7] utilizes external benchmarks only for the analyzed dataset as the clustering is focused on identifying predetermined clusters (ground truth).

4) Internal Indexes: In contrast to *external indexes* the evaluation of goodness of a clustered structure is computed with internal information of the clusters and the dataset itself [7]. The core idea of internal measures is to qualify the notion of having elements of the same cluster close together while having elements of different clusters far apart from each other. In other words, elements of the same cluster are similar, elements of separate clusters are dissimilar. These metrics are useful to gain understanding of how well the clusters are structured in relation to each other. They can be used well to evalute different parameter configurations of the same algorithm as they show the goodness of the resulting clusters. Comparison of the internal indexes between different algorithms needs to be avoided as other assumptions about the cluster structures are made [5].

The most common internal index metric used is SSE. It describes the intra-cluster squared distances. Optimizing for this metric means to look for clusters with more similar elements while disregarding the intra-cluster distance. A metric that is well equipped to optimize both of those components is the Silhouette score which can be based on Euclidean distance. It computes for each point the mean distance to the other elements in the same cluster and the mean nearest cluster distance and sets them into relation to each other [28].

VI. LIMITATIONS

Many clustering algorithms do not work well with time series data. In particular the high dimensionality and noise in time series data pose serious challenges to these algorithms. Aghabozorgi *et al.* [5] point out in their review that the research focused on (1) representing time-series in lower dimensions to make them compatible with conventional clustering algorithms and (2) base the distance measurements on the raw time series data. For (1) those efforts have suffered from the lost information due to dimensionality reduction, for example in the data-adaptive methods. On (2) no clear winner or framework has been proposed to make a decision on which distance measurement to use under which general circumstances. The authors also point out that the common clustering algorithms are suitable for static data which is in contrast to the dynamic nature of time series.

In section II-B 4 different categories of representing time series have been introduced. Considering the studies performed on the data-adaptive and model-based methods it is easy to see that they can help approximate the time series reasonably well and reduce the dimensionality of the data. Using them as representation method becomes challenging when multiple series are to be analyzed. Non-data-adaptive methods struggle when analyzing and clustering multiple time series of varying length.

The second factor impacting the results of clustering time series is the chosen similarity metric. Aghabozorgi *et al.* [5] revealed that the most effective and accurate approaches are based on dynamic programming which is computationally expensive and requires prior fine-tuning. Researchers therefore often turn to well understood and simple to use methods like Euclidean distance and DTW. In part, this is also the case because few metrics exceed the performance of Euclidean distance [25].

The clustering algorithms themselves also have a significant impact on the performance of the clustering results. For example achieving good clustering results with partioning algorithms like K-Medoid or K-Means is very difficult in cases where the cluster shapes are not globular. But in high-dimensional data this property is not easily observable by a user. Outliers also impact the best choice for the prototypes in the named algorithms. Additionally a pre-determined number of clusters also need to be provided and unequal length of time series require adaptations, for example in the representation method of the time series. This makes this algorithm class hard to apply to real world problems. Despite these limitations this algorithm class is the most prevalent in the observed studies. The algorithm categories - hierarchical, model-based, and density-based, address many of the aforementioned challenges for partioning algorithms. This flexibility comes at the price of the computational complexity and the consequence of being primarily useful for small datasets [5].

Most work is focused on uni-variate time series which does not reflect the full spectrum of common time series data. For example in predictive maintenance applications multi-variate time series are prevalent. This area is not well served with the current body of work in time series clustering. Even more fundamentally it is not clear that the requirements for clustering research requested by Keogh and Kasetty [25] in 2003 have been adequately addressed by research that followed. The time series utilized for research is still limited in scope their scope. In fact, this author was not able to find literature that clusters the data from the Makridakis competitions [29].

VII. Conclusions

A enormous amount of research has been conducted in the different areas mentioned in this literature review. In particular clustering, time series representation and distance measures, clustering algorithms, and assessment metrics. From that work no clear pattern has emerged which methods and metrics are to be used in which circumstances. Good results have been shown mostly in limited areas which often lack generalizability to real world problems and applications. Aghabozorgi *et al.* [5] propose to focus research efforts on creating a combination of different algorithms as hybrid or multi-step clustering algorithms.

We have a different perspective. The research conducted so far has devised simple and more complicated approaches and evaluated them without revealing a general set of rules. As pointed out by Keogh and Kasetty [25] many studies lack generalizability. This limits the utility of the created research to narrowly defined areas. However, generalizable results and truths are required if clustering is to serve analytical processes involving time series in a meaningful way.

Adding more complexity into the clustering process by combining algorithms may not serve the goal of achieving better results. Possibly general truths about clustering time series that are argued from first principles would help to remove some complexity from the entire process and allow for a clearer picture when to apply or avoid certain metrics or algorithms. As an example it would be helpful to be able to decide which distance metric is the most meaningful for a given task and not have validate different combinations for various algorithms that are to be analyzed. A framework that defines the order of steps to be carried out and provides what not to do and possibly what do for a given analysis task may be helpful to identify other drivers that exist that influence the performance of time series clustering and how they influence subsequent analytical activities.

The analysis task that is to be executed as the result of a clustering also requires further careful examination. It is not the same to try a find clusters that match a pre-determined classification as done in [7] compared to learning details of a time series contained in a large dataset. The first one has a clear focus on which the target for clustering can be set. The second topic has the aim of finding nuances and peculiarities in a large set of data.

Until steps in the directions described above are made it will continue to require careful design and experimentation by human intervention in a very large solution space to achieve task appropriate time series clustering.

References

[1] V. Estivill-Castro, "Why so many clustering algorithms," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 1, pp. 65–75, Jun. 2002, ISSN: 1931-0153. DOI: 10.1145/568574.568575. [Online]. Available: http://dx.doi.org/10.1145/568574.568575.

[2] P. Tan, M. Steinbach, V. Kumar, and A. Karpatne, *Introduction to Data Mining eBook: Global Edition*. Pearson Education, 2019, ISBN: 9780273775324. [Online]. Available: https://books.google.de/books?id=274oEAAAQBAJ.

[3] C. Antunes and A. Oliveira, "Temporal data mining: An overview," *KDD Workshop on Temporal Data Mining*, pp. 1–13, Jan. 2001.

[4] Y. Tang, Y. Xie, X. Yang, J. Niu, and W. Zhang, "Tensor multi-elastic kernel self-paced learning for time series clustering," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2019, ISSN: 2326-3865. DOI: 10.1109/tkde.2019.2937027. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2019.2937027.

[5] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, "Time-series clustering – a decade review," *Information Systems*, vol. 53, pp. 16–38, Oct. 2015, ISSN: 0306-4379. DOI: 10.1016/j.is.2015.04.007. [Online]. Available: http://dx.doi.org/10.1016/j.is.2015.04.007.

[6] M. Ali, A. Alqahtani, M. W. Jones, and X. Xie, "Clustering and classification for time series data in visual analytics: A survey," *IEEE Access*, vol. 7, no. nil, pp. 181 314–181 338, 2019. DOI: 10.1109/access.2019.2958551. [Online]. Available: https://doi.org/10.1109/access.2019.2958551.

[7] A. Javed, B. S. Lee, and D. M. Rizzo, "A benchmark study on time series clustering," *Machine Learning with Applications*, vol. 1, no. nil, p. 100 001, 2020. DOI: 10.1016/j.mlwa.2020.100001. [Online]. Available: https://doi.org/10.1016/j.mlwa.2020.100001.

[8] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, Feb. 2012, ISSN: 1573-756X. DOI: 10.1007/s10618-012-0250-5. [Online]. Available: http://dx.doi.org/10.1007/s10618-012-0250-5.

[9] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl, "Va2: A visual analytics approach for evaluating visual analytics applications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 61–70, 2016. DOI: 10.1109/TVCG.2015.2467871.

[10] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," vol. 730, Jan. 1993, pp. 69–84.

[11] J. Zhao and L. Itti, "Shapedtw: Shape dynamic time warping," *Pattern Recognition*, vol. 74, pp. 171–184, Feb. 2018, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2017.09.020. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2017.09.020.

[12] A. Mueen and E. Keogh, "Extracting optimal performance from dynamic time warping," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016. DOI: 10.1145/2939672.2945383. [Online]. Available: http://dx.doi.org/10.1145/2939672.2945383.

[13] H. Kamalzadeh, A. Ahmadi, and S. Mansour, "Clustering time-series by a novel slope-based similarity measure considering particle swarm optimization," *Applied Soft Computing*, vol. 96, p. 106 701, Nov. 2020, ISSN: 1568-4946. DOI: 10.1016/j.asoc.2020.106701. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2020.106701.

[14] H. Kamalzadeh, A. Ahmadi, and S. Mansour, "A shape-based adaptive segmentation of time-series using particle swarm optimization," *Information Systems*, vol. 67, pp. 1–18, Jul. 2017, ISSN: 0306-4379. DOI: 10.1016/j.is.2017.03.004. [Online]. Available: http://dx.doi.org/10.1016/j.is.2017.03.004.

[15] W. Abdulla, D. Chow, and G. Sin, "Cross-words reference template for dtw-based speech recognition systems," in *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, vol. 4, 2003, 1576–1579 Vol.4. DOI: 10.1109/TENCON.2003.1273186.

[16] V. Hautamaki, P. Nykanen, and P. Franti, "Time-series clustering by approximate prototypes," *2008 19th International Conference on Pattern Recognition*, Dec. 2008, ISSN: 1051-4651. DOI: 10.1109/icpr.2008.4761105. [Online]. Available: http://dx.doi.org/10.1109/ICPR.2008.4761105.

[17] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35–58, Mar. 2001, ISSN: 0306-4379. DOI: 10.1016/s0306-4379(01)00008-4. [Online]. Available: http://dx.doi.org/10.1016/S0306-4379(01)00008-4.

[18] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999, ISSN: 0018-9162. DOI: 10.1109/2.781637. [Online]. Available: http://dx.doi.org/10.1109/2.781637.

[19] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch," *ACM SIGMOD Record*, vol. 25, no. 2, pp. 103–114, Jun. 1996, ISSN: 0163-5808. DOI: 10.1145/235968.233324. [Online]. Available: http://dx.doi.org/10.1145/235968.233324.

[20] E. Keogh and J. Lin, "Clustering of time-series subsequences is meaningless: Implications for previous and future research," *Knowledge and Information Systems*, vol. 8, no. 2, pp. 154–177, Aug. 2005, ISSN: 0219-3116. DOI: 10.1007/s10115-004-0172-7. [Online]. Available: http://dx.doi.org/10.1007/s10115-004-0172-7.

[21] X. Wang, K. A. Smith, R. Hyndman, and D. Alahakoon, "A scalable method for time series clustering," Tech. Rep., 2004.

[22] T. Warren Liao, "Clustering of time series data—a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, Nov. 2005, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2005.01.025. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2005.01.025.

[23] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wavecluster: A wavelet-based clustering approach for spatial data in very large databases," *The VLDB Journal The International Journal on Very Large Data Bases*, vol. 8, no. 3-4, pp. 289–304, Feb. 2000, ISSN: 0949-877X. DOI: 10.1007/s007780050009. [Online]. Available: http://dx.doi.org/10.1007/s007780050009.

[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "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*, ser. KDD'96, Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[25] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, 2003, ISSN: 1384-5810. DOI: 10.1023/a:1024988512476. [Online]. Available: http://dx.doi.org/10.1023/A:1024988512476.

[26] Z. Hiu, H. Tu Bao, Z. Yang, and L. Mao-Soong, "Unsupervised feature extraction for time series clustering using orthogonal wavelet transform.," *Informatica (03505596)*, vol. 30, no. 3, pp. 305–319, 2006, ISSN: 03505596. [Online]. Available: http://search.ebscohost.

com/login.aspx?direct=true&AuthType=ip,sso&db=iih&AN=23701633&site=eds-live&custid=s1098328.

[27] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2/3, pp. 107–145, 2001, ISSN: 0925-9902. DOI: 10.1023/a:1012801612483. [Online]. Available: http://dx.doi.org/10.1023/A:1012801612483.

[28] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Aug. 2019, ISBN: 1492032646. [Online]. Available: https://www.xarg.org/ref/a/1492032646/.

[29] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, Jan. 2020, ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2019.04.014. [Online]. Available: http://dx.doi.org/10.1016/j.ijforecast.2019.04.014.