Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 93 (2020) 1352–1357

www.elsevier.com/locate/procedia

53rd CIRP Conference on Manufacturing Systems

Comparison of Time Series Clustering Algorithms for Machine State Detection

Martin Hennig[a,*], Manfred Grafinger[a], Detlef Gerhard[b], Stefan Dumss[a], Patrick Rosenberger[a]

[a]*TU Wien Institute of Engineering Design and Product Development – Research Area Mechanical Engineering Informatics and Virtual Product Development, Getreidemarkt 9/307, 1060 Vienna, Austria*

[b]*Ruhr-Universität Bochum Institute of Product and Service Engineering - Chair for Digital Engineering Systems, Universitätsstraße 150, 44801 Bochum, Germany*

* Corresponding author. Tel.: +43-1-58801-30726; fax: +43-1-58801-30798. *E-mail address:* martin.hennig@tuwien.ac.at

Abstract

New developments in domains like mathematics and statistical learning and availability of easy-to-use, often freely accessible software tools offer great potential to transform the manufacturing domain and their grasp on the increased manufacturing data repositories sustainably. One of the most exciting developments is in the area of machine learning. Time series clustering could be utilized in machine state detection which can be used in predictive maintenance or online optimization. This paper presents a comparison of freely available time series clustering algorithms, by applying several combinations of different algorithms to a database of public benchmark technical data.

Keywords: *Machine Learning; Industry 4.0; Time Series Clustering; Predictive Maintenance; Unsupervised Learning; Internet of Things*

1. Motivation

With the ongoing fourth industrial revolution data-driven processes are becoming more relevant for production companies. Recent advances in research and open-use software, made powerful algorithms increasingly available. By applying them to the manufacturing domain, companies can leverage new insights from existing manufacturing data inventories and data sources to further increase production efficiency, flexibility and maintenance scheduling.

In almost every modern industrial application measurements are made over time and stored in an ordered collection of data, called time series (TS). To analyze and structure this otherwise flat and unstructured data, machine learning (ML) algorithms, especially clustering, offer great capabilities [1]. However, the field of ML is very diverse and many different clustering algorithms, theories, and methods are available without sufficient guidance or overview. For various manufacturing practitioners this acts as a barrier regarding the adoption of these powerful tools and thus may hinder a utilization of ML.

Although there are some meta studies that compare TS clustering algorithms, most of them are limited to giving qualitative statements [2] or comparing only a few datasets while establishing a new method [3-5]. None of them compare extensive quantitative results regarding cluster validity and required calculation time which can guide non-ML-experts. Therefore, this work compares TS clustering algorithms to give quantitative statements and recommendations for industrial practitioners. Since there is no property known that is distinctive for TS in industrial applications the comparison is conducted on data from one of the world's largest collection of labelled TS data [6]. To transfer the findings to a manufacturing use case a clustering of machine TS is also conducted in its paper to identify machine states as a use case for the

10.1016/j.procir.2020.03.084

recommended methods application.

Nomenclature

DTW	dynamic time wraping
ED	Euclidian Distance
GMM	Gaussian Mixtures Models
HCTSA	Highly Comparative Time Series Analysis
LCSS	Longest Common Subsequence
m	Length of a queried subsequence or motif
MDL	Minimal Description Length
ML	Machine Learning
n	Length of a time series
STOMP	Scalable Time series Ordered-search Matrix Profile
TS	Time series

A clustering of TS usually comprises a calculation of certain features of a TS, building feature vectors from this, compare the TS's feature vectors to obtain similarity relationships and calculating a clustering from them [2]. Therefore, in order to compare a TS clustering method, the aspect of feature extraction and the clustering algorithm itself have to be considered. Fig. 1 shows the methods which are part of the comparison and their interaction. In the state-of-the-art search various methods for these feature extraction and clustering have been found. They are referenced and reintroduced briefly in chapter 2 and 3. Chapter 4 discusses the used datasets and evaluation indices before the comparison results are presented. The resulting recommendation of a feature extraction and clustering method is then shown in an industrial application in chapter 5 and a conclusion and outlook is given in chapter 6.

The comparison conducted for this paper includes eight feature extraction algorithms combined with eight different clustering algorithms. The clustering results are evaluated with two different performance indices. For this work they have been implemented or adopted in MATLAB by the authors. Due to clustering being an unsupervised procedure, evaluating its performance is subjective. However, a great amount of research has been invested in trying to standardize cluster evaluation metrics by using cluster validity indices and many indices have been developed over the years [7]. For crisp partitions external cluster validity indices can be used if the ground truth is known that may cause the difference in behaviour [2]. In a ML use case in a production environment, this relation is explored and utilized too.

All algorithm combinations have been tested based on a large TS database with real world datasets from various fields. The most accurate and efficient methods were selected and are recommended as a result of the work presented in this paper. The recommendations found here are later used in the implementation of a TS clustering method for a currently unpublished ML methodology to empower process experts to utilize ML in analysis of their process data.

2. Feature Extraction Methods

The comparison carried out for this paper includes eight feature extraction algorithms, which can be categorized by how they find similarity: shape-based similarity and structure-based similarity.

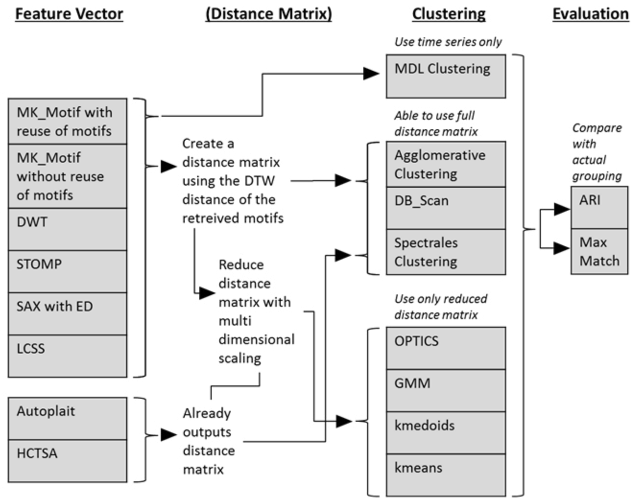

Fig. 1. Illustration showing the different parts of the comparison of the feature extraction and clustering approaches

2.1. Motif Discovery with shape-based similarity

A basic approach to finding similarity in TS is by searching for similar curve shapes in TS. In general, this is done by extracting subsequences of a TS with a sliding window, calculate an accumulated pointwise difference, and looking for the minimum of this value. Similar subsequences have a low accumulated pointwise difference and if they occur repeatedly in similar shapes then are called TS "motifs". Clustering on separated motif parts as candidates instead on all possible subsequences of a certain length is necessary to obtain meaningful results [8]. This is a popular approach to find candidates for a TS clustering and the main difference between the methods lies in the way they calculate the accumulated pointwise difference In terms of TS the abstract meaning of similarity and distance can be transformed into one another by inversion. The DTW distance measure was used because it is more robust than standard Euclidian Distance against small errors and local warping, especially when using less than 10,000 data instances [9]. The calculation of the distance matrix was necessary because two of the feature extraction algorithms do not output motifs but a distance matrix directly and some of the compared clustering methods require it (see Fig. 1).

MK_Motif with and without reuse of motif partners: The Mueen-Keogh Algorithm for Motif Search (MK_Motif) [10] is used to search for motifs using the Euclidian distance. Its main advantage is its calculation time efficiency because the search order is optimized, and early abandoning is used to maximize search space pruning in similarity search.

In this paper MK_Motif is implemented in the original (marked as MK_Motif 1) and a modified version as an experiment (marked as MK_Motif 2) by the authors. The intention is to investigate the effect of reusing motif parts which is not foreseen in the original MK_Motif.

Reusing motifs means that instead of finding a similar pair of TS subsequences and removing both from the search space only one is removed. The subsequence that is more similar to any third subsequence is kept and can be part of another motif which opens up the possibility of similarity chains or tree-like structures. This might have the advantage of creating more structure and relationships in the feature space in which a clustering is conducted.

Symbolic aggregate approximation (SAX): In order to assess whether a lower-bounding combination of similarity measure and representational form offers advantages, SAX was selected for the comparison, which is used together with a Euclidean distance measure and marked within this paper as SAX with ED. SAX is a TS representation, which transforms a TS into a string. The number of time steps that a character represents is fixed and the discretization of the value range is done in a way that produces characters with equiprobability (under the assumption that the TS is normally distributed) [11]. Since SAX is lower-bounding, the exact motif search with early abandoning can be done in the summarized form without having to resort to the complete raw pointwise comparison or risking false-negatives in early abandoning.

STOMP: To compare how a computationally very efficient method performs compared to other methods that work with and without pruning, the Scalable Time series Ordered-search Matrix Profile (STOMP) algorithm was included in the comparison. It solves the problem of TS subsequences All-Pairs-Similarity-Search [12] by calculating the matrix profile and the matrix profile index for the entire TS very efficient. The matrix profile is a meta-time series which gives the distance of each position of a TS to its most similar motif partner of a given length. The matrix profile index is an addition to this, which contains the indices of the exact motifs for each position [12].

STOMP is only one of several algorithms that utilize a matrix profile to perform a TS similarity search exceptionally efficient. It should be mentioned that there are even faster hardware-specialized successor algorithms, and STOMP was used as a general purpose example of a whole group of algorithms which is still in the focus of TS research today [13].

Dynamic Time Wrapping (DTW) and Longest Common Subsequence (LCSS): In order to include similarity measures in the comparison, where time warping or disregard of unassigned TS sections is possible, DTW [14] and LCSS [15] were chosen.

2.2. Motif Discovery with structure -based similarity

Since similarity relationships in TS can not only be determined by shape-based similarity, representatives of structure-based similarity concepts are also included in the comparison. Both methods output a distance matrix directly.

Autoplait: is a method which finds similarity of sections in the structure rather than in the shape [16]. The Autoplait algorithm uses Hidden Markov Models as a representation and matches TS segments as similar that can be approximated by similarly parameterized Hidden Markov Models.

HCTSA: In addition to the similarity assessment of structural similarity, there are also similarities in statistical properties. With the program HCTSA by Fulcher, Little and Jones [17], which arose from a comprehensive literature search on TS parameters, approx. 7,700 statistical parameters are calculated for a TS. These parameters form a basis for creating a high dimensional feature vector which offers many different perspectives on similarity relations.

3. Clustering Methods

Once motifs or features of a TS have been found and a distance matrix is built a general purpose clustering algorithm can be applied. Only Minimal Description Length (MDL) clustering operates directly on a TS. However, a complete distance matrix is usually very high-dimensional, which creates problems for some clustering methods.

Therefore, the distance matrix is reduced to a low-dimensional representation for some methods. A technique known as Multi-Dimensional Scaling is used [18]. It transforms entities with known distance relationships, rotationally invariant in a lower dimensional space so that distances are best preserved.

The direct interpretability of these new centres in the low-dimensional space is no longer given, but this can be recovered, since only their grouping is of interest and the candidates' TS indices remain. Though them it is always possible to find the corresponding time step of a candidate subsequence in the original TS.

3.1. MDL Clustering

The core of this approach is that data is approximated with a model of selectable complexity. To calculate the description length, the information needed to define the model and the one needed to describe the delta of the model's approximation to the real data is summed up. The description length is low if a simple model is found that approximates the real data rather accurate. For a clustering task, the model is typically several cluster cores and a mapping function. A cluster core is the average TS of all the TS that are affiliated with a certain cluster.

3.2. Center-based clustering methods

These methods usually need more points than dimensions for clustering, which is why they require a reduced distance matrix.

Gaussian Mixtures Models: In statistics there is the notion of mixed distributions which can identify subpopulations in a total population. The assumption behind this is that the observed total distribution is the result of a superposition of multiple random variables originating from the same distribution type but with different unknown parameters. In order to be able to estimate the parameters of the subdistributions from the observations as a whole, the expectation-maximization algorithm is used [19]. With this procedure, the parameters of gaussian distribution functions are estimated and can be used as definition for a clustering.

Density-based Clustering Methods (DBSCAN & OPTICS): Since most clustering methods work best on elliptic-shaped clusters, density-based techniques are included in the comparison which are an alternative if clusters have other shapes. Since these do not work directly with distances between objects, but with an object appearance density, they are able to find clusters of any shape. It is also possible to separate clusters of noise or outliers. A well-known density-based clustering method is DBSCAN, which was used as an implementation for MATLAB by Heris [20]. As DBSCAN can have problems with feature distribution, that have varying densities, it's derivative OPTICS was also chosen in the comparison. Here it's MATLAB implementation from Daszykowski [21] is evaluated.

3.3. Agglomerative Clustering and Spectral Clustering

Agglomerative clustering: is a hierarchical clustering method which means it builds a cluster structure that can have multiple sub-clusters. The specific algorithm used in the comparison is "Weighted Pair Group Method with Arithmetic Mean" which is an agglomerative bottom-up procedure for hierarchical cluster formation [22].

Spectral clustering: is an alternative method of cluster analysis which uses a weighted graph to derive a special feature vector and apply a kmeans clustering to it. It uses a form of representation that particularly emphasizes the cluster structure.

Underlying datasets

As there is no distinctive characteristic for TS in industrial applications the authors resorted to finding methods that excel in a broad selection of application fields. The data foundation for the comparison, consists of data from the "UCR TS Classification Archive" [6]. It is one of the largest collections of labeled TS in the world. It contains datasets from technical use cases, medicine, biology, automation, Internet of Things, human motion capture, astronomy and many more. The advantage of using this data is that it has already been used in numerous scientific papers as an evaluation reference which provides a good comparability with other works. Also, the instance labels and true number of clusters are known. To prove usefulness of the results the recommended methods are applied in an industrial use case afterwards in chapter 5.

Data in the archive is one dimensional and structured in datasets which consist of independent instances of TS with a known correct label and a consistent length. To search motifs over a complete dataset the instances were concatenated into one large TS. Therefore, not all positions of this large TS are allowed as motifs, since motifs have to be restricted to a single TS instance. In an industrial setting these TS instances might be production runs or batches being processed at once. This restriction was implemented by the authors into each feature extraction methods script except for STOMP, which does not allow this restriction directly. Therefore, a subordinate script for the evaluation of the matrix profile produced by STOMP to was implemented by the authors to ensure permissibility of the motif positions.

Not all datasets of the collection were used for two reasons: 1) after the first 20 datasets the results did not change in a magnitude that would alter the results. Nevertheless, the calculation was continued for 16 more datasets 2) It was necessary to streamline the comparison because the results as reported in this paper took over 4.5 months of calculation time on a single modern computer workstation.

The authors are aware the industrial TS information is not always one dimensional multidimensional, but these methods are permissible since most of the methods can be applied in a multidimensional manner. Moreover, is this set of methods is part of an ongoing development of a larger ML methodology in which the TS clustering is preceded by an extensive prepossessing which performs data cleaning and dimensionality reduction to mostly univariate TS.

4. Results and Interpretation

Evaluation

For the evaluation of the clustering results two indices were used to assess the permutation-corrected matching of the true and estimated grouping. The first was the Adjusted Rand Index [23]. It is 1 in case of a perfect match, 0 for random guessing and negative if worse. Adjusted means that it considers the differences in expected matches for clusters of unequal sizes. The second evaluation index was the symmetric Maximum-Match-Measure [24]. It iteratively matches clusters 1: 1 based on the maximum value of the confusion matrix.

Since the indices have different ranges and datasets have varying "difficulties" reporting the absolute sums of the indices' values would not be a robust evaluation. Instead the evaluation is based on the inverse ranks as scores (Higher ranks mean a better result). The inverse ranks were summed up over all processed datasets and all evaluation indices.

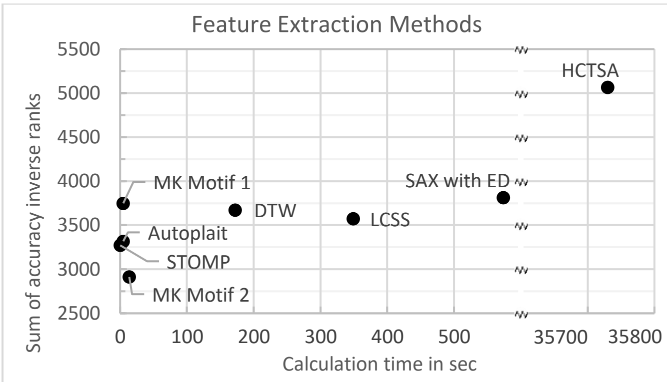

Fig. 2. Accuracy vs. calculation time for feature extraction methods

As seen in Fig. 2 the feature extraction method which produced the best accuracy over all clustering methods is HCTSA but it is needs 60x more calculation time than the next best method. Because feature extraction is by far the most resource intense part of TS clustering, one should rather use a motif search MK_Motif 1 for this task if calculation time is an issue.

Although MK_Motif 1 is faster than STOMP in this comparison, an unimpaired Matrix Profile algorithm that is allowed to use all positions of the search space will be faster in most cases [13]. Since all three will generate exact motifs, the expected accuracy should be close to same. Accuracy differences in this comparison are due to the script that restrains

STOMP or due to rounding within the calculation of ranks as no double ranking was allowed.

It becomes obvious that reusing motifs in MK_Motif 2 did not help in the accuracy of the clustering, so a practitioner should resort to the original Mueen-Keogh motif algorithm. Presumably this happened because motifs found with MK_Motif 1 are more independent of each other and have more discrimination potential as they are more distinct which improves their potential as a clustering feature.

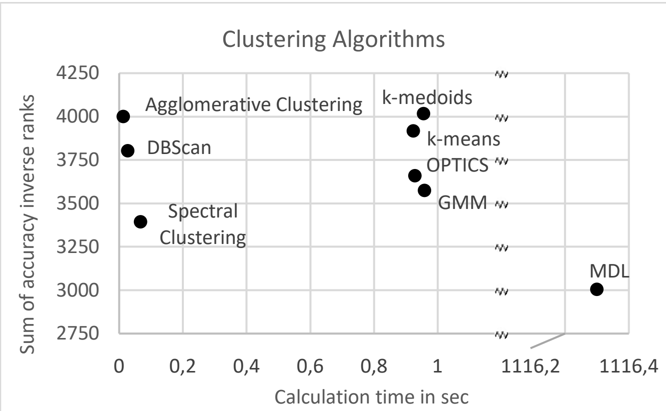

Fig. 3. Accuracy vs. calculation time for the compared clustering methods

Fig. 3 shows that agglomerative clustering and k-medoids scored higher accuracy ranks more often than others. But it needs to be said that all methods results are close together, except for MDL Clustering. Likely MDL scored worse because it requires to find the correct number of clusters itself instead of all others which take this information as a parameter. This limits the comparability but if this parameter is not given this method can be a practical option. If the cluster number had to be guessed otherwise there are separate algorithms for that, which can be very resource intensive too.

Since the database is used by other researchers the results can be compared to those of other works. For example Tamura and Ichimura [4] tested a novel distance measure based on SAX, compared it to other distance measures and reported its rand index in clustering on this database using k-medoids. Their results scored on average 10% higher than the values of this paper which might be due to the differences between rand index and adjusted rand index. Zhu, Batista, Rakthanmanon and Keogh [5] use the UCR archive data to test clustering performances of a novel distance measure and Lin, Vlachos, Keogh and Gunopulos [3] use it to evaluate an incremental clustering method. Both are limited to only selected datasets of the archive. For the few datasets which were used there and in this work the results could be confirmed.

Interpretation

As a result of the comparison the following recommendation for a TS clustering task is given: If calculation time is not an issue, then HCTSA should be used to find a distance matrix. Otherwise for faster calculation either STOMP or the original MK_Motif should be used depending on whether all TS positions are allowed as motifs or not. For clustering of the distance matrix either k-medoids or an agglomerative clustering should be used.

Once typical clusters of a production line's TS are known they can be interpreted and associated with machine states or products or operation mode depending on what shows significance or what is of interest. For this clusters can be altered (e.g. split/merged/shifted) in an interpretation by a knowledgeable process expert as a significant grouping in terms of a feature vector must not necessarily have a relevant meaning in terms of a production process.

If the clusters are interpreted and assigned to known relevant machine states, they can be used for TS classification or further data-based automation. For example, current or past machine TS could be classified to machine states and hence enriched with meta-information. This could be used to indicate a certain wear (to support a predictive maintenance strategy), structure an otherwise flat TS, visualize machine activity or ease further information processing like adjusting data aggregation or training neural networks.

5. Use Case Machine State assignment

Fig. 4 shows a set of 1,000 already preprocessed univariate anonymized movement measurements taken from an ongoing research project at the TU Wien pilot factory Industry 4.0 [25] about robotic arm degradation. In Fig. 4 it may be visible that there are two groups of curve shapes. The upper corresponds to the initial state and the lower to critical levels of degradation. But since the degradation only results in a variance in the curve that is lower than the overall curve's variance it is hard to detect the onset of degradation before it is severe. With ML based machine state detection that uses the recommended methods this becomes possible.

At first a high dimensional feature vector is built for each measurement dataset and transformed into a distance matrix by HCTSA. No parameters had to be set, and standard settings were used. The distance matrix was then clustered to the desired number of clusters (four was chosen) with "Weighted Pair Group Method with Arithmetic Mean" linkage agglomerative clustering. This resulted in the four curves shown in Fig. 5 that represent the cluster cores as a mean of each cluster. Upon consultation with the researchers at the pilot factory the clusters could be linked with an onset, and an advancing of the robot arm degradation. In the future a new measurement can be compared to these four cluster cores and if a similarity is detected then a degradation level can be assigned. Thanks to the hierarchical character of the agglomerative clustering different sub-clusters to each degradation level can be identified (by using more clusters) which can be used to help in a detailed investigation in a fault analysis. The calculation of HCTSA on all measurements took 72 minutes. The curves which belong to a critical degradation were all put in the same cluster with a Maximum-Match-Measure of 0.98, the initial level curves were clustered together with a Maximum-Match-Measure of 0.93.

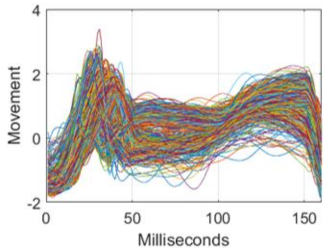

Fig. 4. All Measurements of the industrial use case

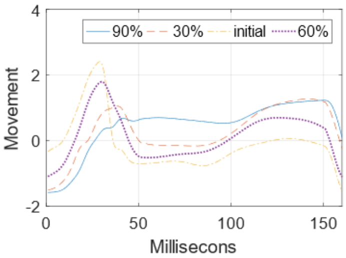

Fig. 5. Identified clusters

In the comparison calculation and the use case each instance corresponds to a meaningful label or state. In a general industrial use case, the data might have instances which cannot be assigned a meaningful label as they might be outliers, transition states or the measurement was corrupted beyond correction. In these cases, it makes sense to provide a label or cluster for outliers and otherwise unclustered instances. [26] If a motif search is used for feature extraction the maximum number of extracted motifs should be limited so that not all possible TS subsequences have to become a candidate.

6. Conclusion and Outlook

This paper presented and compared different feature extraction and clustering methods in order to recommend practitioners algorithms for TS clustering. The utilization of this recommendation for machine state detection was showcased with an industrial real-world example.

It was found that it is advisable to use HTCSA as feature extraction and k-medoids for TS clustering. If the application calls for a fast calculation feature extraction with MK_Motif or a Matrix profile algorithm like STOMP are faster alternatives.

With these methods it is possible to form clusters from machine TS. These can then be easily interpreted and assigned to meaningful machine states which opens a plethora of potential uses.

However, in practice some aspects which were given here, due to space limitations, are not easy to achieve. For example, a procedure that dictates how TS instances are formed or how outlier cleaning should function is often complex and depended on purpose. But these tasks are topic of many existing research works and are definitely feasible. [27]

References

[1] Tu Q, Vonderembse M A, Ragu-Nathan T, The impact of time-based manufacturing practices on mass customization and value to customer, Journal of Operations management, 19 (2001) 201-217.

[2] Warren Liao T, Clustering of time series data—a survey, Pattern Recognition, 38 (2005) 1857-1874.

[3] Lin J, Vlachos M, Keogh E, Gunopulos D, Iterative incremental clustering of time series, International Conference on Extending Database Technology, Springer, 2004, pp. 106-122.

[4] Tamura K, Ichimura T, Clustering of time series using hybrid symbolic aggregate approximation, IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2017, pp. 1-8.

[5] Zhu Q, Batista G, Rakthanmanon T, Keogh E, A novel approximation to dynamic time warping allows anytime clustering of massive time series datasets, Proceedings of the 2012 SIAM international conference on data mining, SIAM, 2012, pp. 999-1010.

[6] Dau H A, Keogh E, Kamgar K, Yeh C-C M, Zhu Y, Gharghabi S, Ratanamahatana C, Ann Y, Hu B, Begum N, Bagnall A, Mueen A, Batista G, Hexagon-ML, The UCR Time Series Classification Archive, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/, 2018.

[7] Arbelaitz O, Gurrutxaga I, Muguerza J, PéRez J M, Perona I, An extensive comparative study of cluster validity indices, Pattern Recognition, 46 (2013) 243-256.

[8] Keogh E, Lin J, Truppel W, Clustering of time-series subsequences is meaningless: implications for previous and future research, Third IEEE International Conference on Data Mining, IEEE, 2003, pp. 115-122.

[9] Wang X, Mueen A, Ding H, Trajcevski G, Keogh E, Experimental comparison of representation methods and distance measures for time series data, Data Mining and Knowledge Discovery, 26 (2012) 275-309.

[10] Mueen A, Keogh E, Zhu Q, Cash S, Westover B, Exact Discovery of Time Series Motifs, (2009) 473-484.

[11] Lin J, Keogh E, Wei L, Lonardi S, Experiencing SAX: a novel symbolic representation of time series, Data Mining and knowledge discovery, 15 (2007) 107-144.

[12] Yeh C-C M, Zhu Y, Ulanova L, Begum N, Ding Y, Dau H A, Silva D F, Mueen A, Keogh E, Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets, IEEE ICDM, 2016.

[13] Chin-Chia Michael Y, Yan Z, Liudmila U, Nurjahan B, Yifei D, Hoang Anh D, Diego Furtado S, Abdullah M, Eamonn K, UCR Matrix Profile Page,

[14] Keogh E, Exact indexing of dynamic time warping, Proceedings of the 28th international conference on Very Large Data Bases, VLDB Endowment, Hong Kong, China, 2002, pp. 406-417.

[15] Vlachos M, Hadjieleftheriou M, Gunopulos D, Keogh E, Indexing multi-dimensional time-series with support for multiple distance measures, Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 216-225.

[16] Matsubara Y, Sakurai Y, Faloutsos C, Autoplait: Automatic mining of co-evolving time sequences, Proceedings of the 2014 ACM SIGMOD international conference on Management of data, ACM, 2014, pp. 193-204.

[17] Fulcher B D, Little M A, Jones N S, Highly comparative time-series analysis: the empirical structure of time series and their methods, Journal of The Royal Society Interface, 10 (2013) 20130048.

[18] Cox T F, Cox M A, Multidimensional scaling, Chapman and hall/CRC, 2000.

[19] Moon T K, The expectation-maximization algorithm, Ieee Signal Proc Mag, 13 (1996) 47-60.

[20] Heris S M K, Implementation of DBSCAN Clustering in MATLAB, 2018.

[21] Daszykowski M, Matlab code for OPTICS, 2018.

[22] Sokal R R, Michener C, A statistical method for evaluating systematic relationships, University of Kansas, Science Bulletin, 38 (1958) 1409-1438.

[23] Hubert L, Arabie P, Comparing partitions, Journal of classification, 2 (1985) 193-218.

[24] Wagner S, Wagner D, Comparing clusterings: an overview, Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.

[25] Hennig M, Reisinger G, Trautner T, Hold P, Gerhard D, Mazak A, TU Wien pilot factory Industry 4.0, Procedia Manufacturing, 31 (2019) 200-205.

[26] Rakthanmanon T, Keogh E J, Lonardi S, Evans S, Time series epenthesis: Clustering time series streams requires ignoring some data, 2011 IEEE 11th International Conference on Data Mining, IEEE, 2011, pp. 547-556.

[27] Dumss S, Grafinger M, Hennig M, Rosenberger P, Combining multiple data sources and enriching the dataset using industrial edge devices, 53rd CIRP Conference on Manufacturing Systems, Procedia CIRP, 2020.