K-MDTSC: K-Multi-Dimensional Time-Series Clustering Algorithm

electronics

MDPI

Article

K-MDTSC: K-Multi-Dimensional Time-Series Clustering Algorithm

Danilo Giordano [1,*,†], Marco Mellia [2,†] and Tania Cerquitelli [1,†]

[1] Department of Control and Computer Engineering, Politecnico di Torino, Corso Duca degli Abruzzi, 24-10129 Turin, Italy; tania.cerquitelli@polito.it
[2] Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi, 24-10129 Turin, Italy; marco.mellia@polito.it
* Correspondence: danilo.giordano@polito.it; Tel.: +39-011-090-7171
† These authors contributed equally to this work.

Abstract: The increasing capability to collect data gives us the possibility to collect a massive amount of heterogeneous data. Among the heterogeneous data available, time-series represents a mother lode of information yet to be fully explored. Current data mining techniques have several shortcomings while analyzing time-series, especially when more than one time-series, i.e., multi-dimensional time-series, should be analyzed together to extract knowledge from the data. In this context, we present *K-MDTSC* (K-Multi-Dimensional Time-Series Clustering), a novel clustering algorithm specifically designed to deal with multi-dimensional time-series. Firstly, we demonstrate *K-MDTSC* capability to group multi-dimensional time-series using synthetic datasets. We compare *K-MDTSC* results with *k-Shape*, a state-of-art time-series clustering algorithm based on K-means. Our results show both *K-MDTSC* and *k-Shape* create good clustering results. However, *K-MDTSC* outperforms *k-Shape* when complicating the synthetic dataset. Secondly, we apply *K-MDTSC* in a real case scenario where we are asked to replace a scheduled maintenance with a predictive approach. To this end, we create a generalized pipeline to process data from a real industrial plant welding process. We apply *K-MDTSC* to create clusters of weldings based on their welding shape. Our results show that *K-MDTSC* identifies different welding profiles, but that the aging of the electrode does not negatively impact the welding process.

Keywords: machine learning; clustering; predictive maintenance; industry 4.0

Citation: Giordano, D.; Mellia, M.; Cerquitelli, T. K-MDTSC: K-Multi-Dimensional Time-Series Clustering Algorithm. *Electronics* **2021**, *10*, 1166. https://doi.org/10.3390/electronics10101166

Academic Editor: Rashid Mehmood

Received: 12 April 2021
Accepted: 12 May 2021
Published: 13 May 2021

1. Introduction

Thanks to the Internet of Things technology and diversified sensors nowadays, we can collect a huge amount of heterogeneous data. Among the heterogeneous data available, time-series are commonly found in different fields such as sales, stock prices, weather, biomedical measurements, industrial data, audio, and video signals, etc. Time-series collect several subsequent observations of the same sensors, thus creating a large set of ordered samples, possibly having multiple dimensions too. Time-series data fostered many studies in the data mining community to promote novel machine learning algorithms and applications: from time-series forecasting in stock trading [1] or sensors networks [2], to classification time-series for health or video data [3], up to clustering to help the Decision Support Systems while analyzing robots trajectories data in rescue operations [4] to name a few. Frequently, researchers rely on time-series transformations to reduce it or its multi-dimensional nature to solve the course of dimensionality problem. Different distance metrics are used to compute the similarities among the transformed time series [5]. Considering the nature of the data, uni-dimensional time-series involve the analysis of one single time-series per time. However, in many fields, the need to analyze multiple and time-series simultaneously is rising, i.e., multi-dimensional time-series. The industrial field is one of them. With the Industry 4.0 paradigm, we can implement robots with multiple

sensors to collect and store a massive amount of data about the manufacturing processes. These robots generate multiple time-series, synchronized over time.

In this paper, we focus on this scenario where synchronous multi-dimensional time-series are available. Firstly, we propose *K-MDTSC* (*K-Multi-Dimensional Time-Series Clustering*), a novel clustering algorithm specifically designed to deal with synchronous multi-dimensional time-series. Based on a generalized version of K-means [6] and a generalized notion of distance able to handle synchronous multi-dimensional time-series of any dimension, *K-MDTSC* automatically groups multi-dimensional time-series. In a nutshell, the generalized distance and the synchronous nature of the data allow *K-MDTSC* creates clusters without the need for any transformation or dimensionality reduction. We compare *K-MDTSC* performance with *k-Shape* [7] a state-of-art time-series clustering algorithm based on K-means. We run a thorough validation process using synthetic datasets composed of distinct synchronous multi-dimensional time-series *families*. We study the impact of the main multi-dimensional time-series data characteristics, i.e., we inject noise-like perturbation, increase the number of families and the number of dimensions. Our results show that both *K-MDTSC* and *k-Shape* effectively identify and group multi-dimensional time-series of the same family. However, *K-MDTSC* outperforms *k-Shape* when noise increases, a common problem in real data, and families' numbers grow.

Our second goal is to apply *K-MDTSC* in a real case scenario. In detail, we employ *K-MDTSC* in an industrial case study to gain a more profound knowledge of the process and understand if a predictive maintenance strategy can be applied instead of the existing periodic one. To this end, we design a generic pipeline to process the data and fed them into our clustering algorithm. Firstly, in our pipeline, we reduce the eventual presence of noise in each dimension of the multi-dimensional time-series employing a low pass filter. Next, we use two data selection processes to align the multi-dimensional time-series obtaining synchronous time-series having equal duration. Finally, different dimensions may be characterized by time-series having different operational ranges; thus, we run a data normalization process. We employ our generic pipeline and *K-MDTSC* on a real dataset composed of two-dimensional time-series collected in the body-in-white welding stage of an actual car manufacturing plant. During this stage, different car bodies' welding spots are soldered together to join the different parts of the car frame. Our results show that the automatic pipeline effectively processes the data so that the *K-MDTSC* algorithm clearly identifies different welding profiles. Furthermore, we show that there is no negative impact on the welding profiles, suggesting that an increase in the wear and tear of the electrode would be possible.

To support new analyses in similar scenarios where synchronous multi-dimensional time-series are available, we release our synthetic datasets and *K-MDTSC* code as open-source at [8].

The contributions of this work can be summarized as follows:

- Definition of *K-MDTSC* (*K-Multi-Dimensional Time-Series Clustering*) to group multi-dimensional time-series.
- Thorough validation of *K-MDTSC* and comparison with *k-Shape*, a state-of-art time-series clustering algorithm.
- Synthetic datasets and *K-MDTSC* code available at [8].
- Generic pipeline to process time-series data.
- Application in a real case scenario with real data from the Industrial field.

The rest of the paper is organized as follows: In Section 2, we discuss our work in light of past literature for clustering algorithms, predictive maintenance, and applications of clustering algorithms with a focus on time-series data. After, we report the details about *K-MDTSC* and *k-Shape* clustering algorithms in Section 3, and present our validation results in Section 4. Next, we present our real case scenario, the available dataset, and detail the steps of our automatic processing pipeline in Section 5. Then, we present the relative clustering results in Section 6. Finally, in Section 7, we draw our conclusions and derive our future directions.

2. Related Work

With the increasing amount of unlabeled data, clustering algorithms have been widely used in the last years to extract knowledge from data in different fields such as image processing [9–12], text analysis [13,14], network traffic analysis [15–18], social network data analysis [19–21], and medical data [22–24]. Along with unlabeled data, the possibility to collect data over time leads to the acquisition of a considerable amount of time-series data. To cluster time-series, authors in [7] proposed *k-Shape*, a K-means-based clustering algorithm to group uni-dimensional time-series by using a dynamic time warping distance. Like them, authors in [25] used a dynamic time warping approach focusing mainly on how to compute the clusters' centroid. Authors in [26], instead, studied how to use a partitional clustering algorithm to handle multi-dimensional time-series in case of both categorical and continuous values. Authors in [27] studied how to rely only on part of the time-series to extract local patterns and use them to cluster time-series having different lengths. Authors in [28], instead, proposed a clustering algorithm to group sub-sequences in multi-dimensional time-series. To study how to transform time-series data into different domains, authors in [29] used 38 distinct datasets and evaluated different time-series representations and different similarity measures to assess the pruning power of each solution. Authors in [30], instead, employed an iterative discrete Wavelet Transformation decomposition with a standard K-means algorithm to group time-series. Authors in [31] used a dimensional reduction solution to replace time-series data with global measures describing the time-series characteristics, and authors in [32] used a Principal Component Analysis methodology to segment and cluster multi-dimensional time-series. Regarding the applications, many works applied clustering to time-series: starting from the financial field [33–35], in which time-series clustering have been used to find seasonal patterns or expected return-risk pattern; with urban data [36] to analyze time-series about population distribution during a day detecting a similar pattern in population distribution changes in a different location; by using environmental data [37] to identify climate indices and group weather stations located in the same state; focusing on meteorological data [38] to propose a methodology able to group multi-dimensional time-series up to the medical field [39,40] in which cluster of biomedical images have been created for several different medical applications from human brain mapping up to the analysis of suspicious lesions in patients with breast cancer. To give a broad view of literature regarding time-series clustering, in 2005, Liao [41] presented a seminal survey regarding time-series clustering. He summarized the previous studies related to time-series clustering, identifying similarities between time-series, and the employed performance metrics. To update it, in 2015, authors in [5] proposed a new overview of ten years of time-series clustering applied in different fields. While recently, in 2020, authors in [42] benchmark eight different clustering solutions with different distance metrics on over 100 time-series datasets. The increasing amount of data available also fostered an increasing interest in predict maintenance approaches in different fields such as railway [43], electric distribution network [44], aircraft [45], industrial vehicles [46], automotive [47,48], and electrical motors [49] to name a few, as well as in robotics industry [50,51], similarly to our case study. Regarding the application of time-series analysis for predictive maintenance, authors in [52] proposed a predictive maintenance pipeline based on clustering to study the laser melting manufacturing technology. Their study proposed a preprocessing step to transform time-series into statistical features, e.g., minimum, skewness, maximum, mode, etc. Similarly, authors in [53] propose the adoption of custom features and the application of a Principal Component Analysis process to select the most important features. Kulkarni et al. [54,55] use clustering to find outliers and extract features, and then use them with autoregressive or classification models to predict faults in industrial and refrigerator machines. Similarly, authors in [56] employed an iterative K-means solution in the data preprocessing step to reduce the signal to analyze in the final classification step.

The closest work to ours is proposed by authors in [57]. In this work, the authors proposed a methodology to group multi-dimensional time-series using a spectral clustering

algorithm. Then, they validated it and compared the clustering performance with a standard K-means implementation employing a synthetic and a networking dataset. Unlike the previous works, in our clustering solution, we do not deal with sub-sequences or time-series re-alignment, as we assume synchronous multi-dimensional time-series having the same length. Moreover, we do not employ any transformation technique to transform time-series data into features or a different domain, e.g., Fourier transforms, wavelet transformation, etc. Instead, we propose a generic multi-dimensional time-series clustering algorithm directly relying on raw synchronous multi-dimensional time-series based on a generalized notion of the euclidean distance. We validate it by comparing the clustering performance with another state-of-art time-series clustering algorithm, i.e., *k-Shape*.

Furthermore, we propose a generic pipeline to preprocess multi-dimensional time-series and fed them into our generic multi-dimensional time-series clustering algorithm. To the best of our knowledge, we are the first to study a real multi-dimensional time-series dataset from the industrial field and suggest using this clustering approach to perform a predictive maintenance strategy.

3. Time-Series Clustering

This section recalls the K-means clustering algorithm that we use as a building block in *K-MDTSC*. Then, we present *k-Shape* as a state-of-art time-series clustering algorithm.

3.1. K-Means

K-means is a classic clustering algorithm born from statistics. It creates central-based clusters such as the points within a cluster are closer (hence more similar) to the centroid of the cluster (i.e., the center of the cluster) they belong to, rather than the centroid of the other clusters. [6]. In K-means, the user specifies a single parameter k, representing the number of desired clusters. Then, starting from the input points, K-means groups them in k clusters, assigning them to the closest centroid. Then, it returns each cluster and the respective centroid.

Initially, K-means randomly extracts k points in the input data space and uses them as initial centroids of the clusters. All the input points are then assigned to the cluster having the shortest distance (usually the Euclidean distance) from the respective centroids. Once K-means assign all the points to a cluster, new centroids are computed and compared with the previous ones. If the centroids do not change, the algorithm stops and returns the generated clusters and centroids. Otherwise, the algorithm restarts reassigning all the points to the clusters based on the new centroids. While traditional K-means represents a simple yet efficient algorithm to group points together, it has some well-known limitations related to the definition of distance and some well-known criticalities, such as creating empty clusters. Most of all, K-means cannot easily deal with time-series.

3.2. k-Shape

k-Shape is one of the state-of-art time-series clustering algorithms based on K-means [7]. To cope with time-series, *k-Shape* employs a *shape-based* distance to evaluate the similarity between two curves. In addition, the *shape-based* distance uses a *cross-correlation* distance to identify the smallest distance between two curves even if they are not properly aligned. To this end, firstly, it shifts one of them to identify the best alignment leading to the smallest distance. Then to cope with time-series inherent distortions, *k-Shape* uses a z-normalization process. Finally, *k-Shape* computes the *shape-based* distance by normalizing the cross-correlation distance by the geometric mean of the autocorrelations of the individual sequences.

While *k-Shape* can identify time-series clusters even when they are not aligned, it cannot handle multi-dimensional time-series per se. Indeed, *k-Shape* gets as input only one-dimension time-series. Here, we adapt it to multi-dimensional time-series to cope with this constraint.

Given a multi-dimensional time-series $X_N(z)$, where N represent the dimensions, we define $X(z)$ as a one-dimension time-series by concatenating all the dimensions as follows:

$$X(z) = \bigcup_{n=0}^{N} X_n(z)$$

Finally, we give as input the $X(z)$ time-series to *k-Shape*.

3.3. K-MDTSC

For our implementation, we base *K-MDTSC* on the traditional K-means algorithm. Firstly, we define a generalized notion of distance to handle time-series, and in particular multi-dimensional time-series.

Given a pair of multi-dimensional time-series $X_N(z)$ and $Y_N(z)$, where z represents the sample in Z samples, and N the dimensions, we define our generalized distance as follow:

$$d(X_N, Y_N) = \sqrt[L]{\sum_{n=0}^{N}\sum_{z=0}^{Z} |X_n(z) - Y_n(z)|^L}$$

where L represents the metric distance. For our implementation, we rely on $L = 2$, i.e., the Euclidean distance. We use the distance $d(.)$ to find the closest centroid in the K-means algorithm. Notice that our generalized distance assumes that $X_N(z)$ and $Y_N(z)$ are synchronous multi-dimensional time-series. In case the multi-dimensional time-series may present different phases but the same shape, i.e., synchronous but not aligned time-series, in Section 5.2, we present our generic pipeline to align them.

In addition, in our implementation, we introduce the following improvements to the classic K-means algorithm:

- A well-known criticality is related to the choice of the initial centroids. If a random point is chosen as a centroid and lays in part of space where no other points belong, it may create empty clusters. The worst the clustering result is, in the end, instead of k clusters, the user will retrieve $k - 1$ clusters. To reduce this situation probability, we choose the k initial centroids among the multi-dimensional time-series directly.
- Secondly, while the first solution reduces the probability of getting empty clusters, some corner cases are still possible, e.g., if two similar multi-dimensional time-series are present and both are used as centroids. To solve this issue, we verify whether any empty cluster exists. If any is present, for each empty cluster, we select from the most prominent cluster the farthest multi-dimensional time-series and take it as a new centroid.

3.4. Performance Metrics

We assess the clustering performance through quality metrics, including the Error Sum of Squares (SSE) and the Adjusted Rand index.

Error Sum of Squares (SSE): The SSE index computes the cluster cohesion [58]. On the one hand, when the points belonging to a cluster lay close to the cluster's centroid, the cluster is very compact; hence a good clustering result is achieved. On the other hand, when points are far from their cluster's centroid, the result is bad, as it means that the points have little similarities. Thus, given a cluster $C \in \hat{C}$ clusters, its multi-dimensional time-series centroid $C_N(z)$ and the time-series $X_N(z) \in C$ on N dimensions, we compute the SSE index as follow:

$$SSE = \sum_{C \in \hat{C}} \sum_{X_N \in C} d(X_N, C_N)$$

Adjusted Rand index: The Adjusted Rand Index [59] measures the clustering algorithm's ability to separate points belonging to preassigned categories and re-identify them without the category's notion. This index is bounded between 0 and 1. When the clustering result

and the preassigned categories perfectly agree, the Adjusted Rand index is 1. Otherwise if, for random labeling, it gets a value close to 0. Since the Adjusted Rand Index is an external index, hence independent from the clustering algorithm and the distance matrix, we rely on the implementation defined in [60].

4. Results on Synthetic Data

In this section, we present our synthetic dataset, and we evaluate the capability of *k-Shape* and *K-MDTSC* to identify groups of homogeneous curves correctly.

4.1. Synthetic Dataset Generation

We generate synthetic datasets, composed of synchronous multi-dimensional time-series by using as building blocks the following functions:

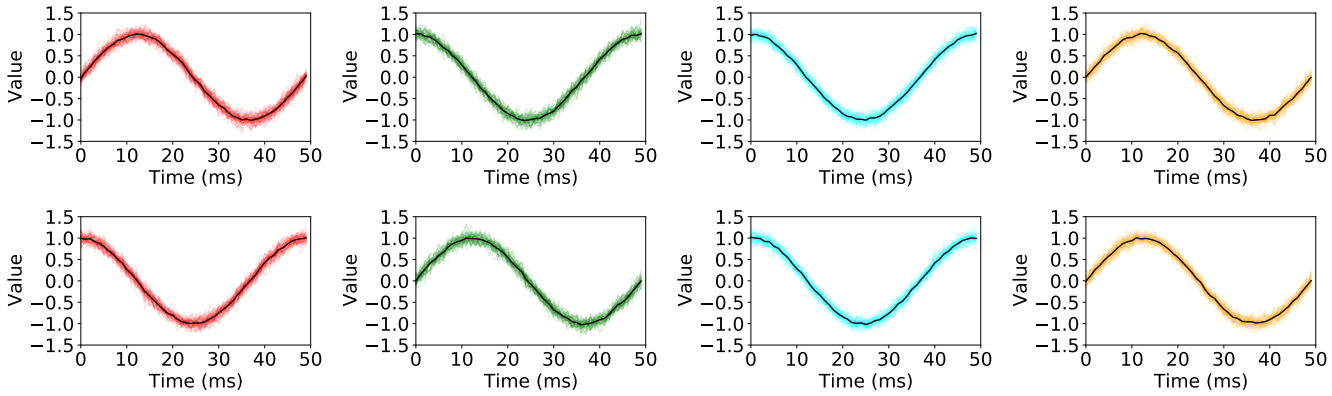
$$Building\ Block = \begin{cases} X(t) = sin(2\pi ft) + N_\sigma(t) \\ Y(t) = cos(2\pi ft) + N_\sigma(t) \end{cases}$$

where f is the frequency of the sinusoidal curve, $N_\sigma(t)$ is a random normal noise component with zero mean and standard deviation σ. We add to each curve this perturbation to create time-series having different profiles.

Based on that, we create datasets composed of multi-dimensional time-series. Firstly, we generate a dataset composed of 200 two-dimension time-series, divided in four distinct families. We set $f = 1$ and $\sigma = 0.1$. In detail, we generate the following families F:

$$Dataset = \begin{cases} F_0 : (X, Y), f = 1, \sigma = 0.1 \\ F_1 : (Y, X), f = 1, \sigma = 0.1 \\ F_2 : (X, X), f = 1, \sigma = 0.1 \\ F_3 : (Y, Y), f = 1, \sigma = 0.1 \end{cases} \quad (1)$$

Figure 1 depicts the 4 two-dimension time-series families present in our dataset. Each family F is composed of 50 synchronous two-dimension time-series. From the plots, we can see how each family presents synchronous time-series, each having slightly different profiles, as shown by the resulting thickness of each family.



(**a**) $F_0 : X, Y, \sigma = 0.1, f = 1$ (**b**) $F_1 : Y, X, \sigma = 0.1, f = 1$ (**c**) $F_2 : X, X, \sigma = 0.1, f = 1$ (**d**) $F_3 : Y, Y, \sigma = 0.1, f = 1$

Figure 1. Synthetic dataset with four time-series families in two dimensions. Black curves report the centroids according to the *K-Multi-Dimensional Time-Series Clustering* algorithm.

We then run the *k-Shape* and *K-MDTSC* on all samples. We test different values of k to identify whether we achieve the best clustering performance when k is equal to the number of families. Figure 2 reports the SSE score while increasing the number of clusters k. As expected, both algorithms identify the best result starting from $k = 4$. To characterize

the clustering results, the black curves in Figure 1 report the centroid of each family as returned by *K-MDTSC*. Similar results hold for *k-Shape*.

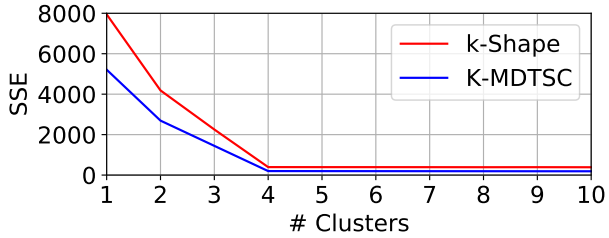
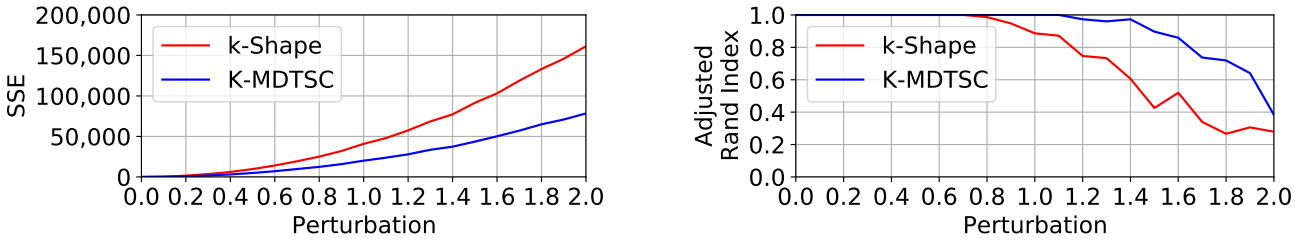


Figure 2. SSE Score of the synthetic data.

Perturbation Stability: Next, we study the clustering capability to handle a growing perturbation. To this end, we generate datasets composed again of 4 distinct synchronous families of two-dimension time-series. We generate datasets having always the same frequency $f = 1$, and characterized by a growing perturbation $\sigma \in [0, 2]$ with step 0.1.

$$Datasets = \begin{cases} F_0 : (X, Y), f = 1, \sigma \in [0, 2] \\ F_1 : (Y, X), f = 1, \sigma \in [0, 2] \\ F_2 : (X, X), f = 1, \sigma \in [0, 2] \\ F_3 : (Y, Y), f = 1, \sigma \in [0, 2] \end{cases} \tag{2}$$

For each dataset, we run the clustering algorithms fixing the parameter $k = 4$ as the number of families. Figure 3a reports the trend of the SSE score. As expected, the higher the perturbation, the higher the SSE score, with a $SSE = 0$ when no-perturbation is present. Interestingly, *K-MDTSC* creates more cohesive clusters and better centroids, as shown by the lower SSE score. To evaluate the capability of the clustering algorithm to automatically assign points of the same family to the same cluster, in Figure 3 we compute the Adjusted Rand Index. We can use it in our synthetic dataset as we are aware of each multi-dimensional time-series family. With $\sigma \leq 1$, both algorithms create a distinct cluster for each family. With a growing perturbation, *k-Shape* creates more heterogeneous clusters composed of time-series belonging to different families than *K-MDTSC*.



(**a**) SSE score (**b**) Adjusted Rand Index

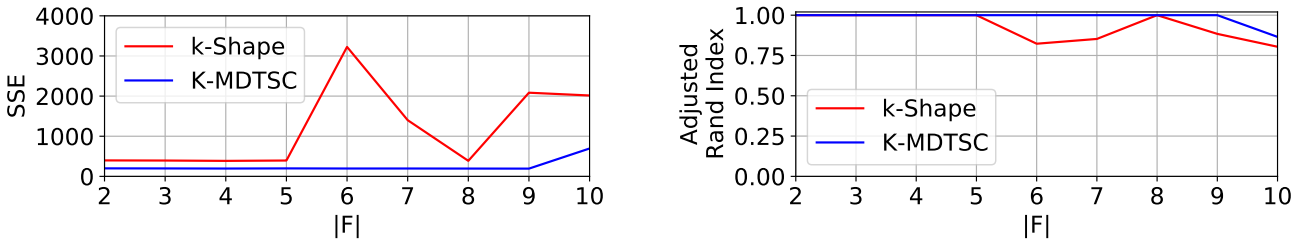
Figure 3. Perturbation Stability.

Families Stability: Next, we study the stability while increasing the number of families. To this end, we generate datasets composed of $|F|$ distinct families of two-dimension time-series. We generate datasets with $|F| \in [2, 10]$. We characterize each family with a unique combination of sinusoidal curves and sinusoidal periods. As the initial dataset, we keep $\sigma = 0.1$.

$$Datasets = \begin{cases} F_{0,4,8} : (X,Y), f \in \{1,2,3\}, \sigma = 0.1 \\ F_{1,5,9} : (Y,X), f \in \{1,2,3\}, \sigma = 0.1 \\ F_{2,6} : (X,X), f \in \{1,2\}, \sigma = 0.1 \\ F_{3,7} : (Y,Y), f \in \{1,2\}, \sigma = 0.1 \end{cases}$$

where the families F_i $i > 4$ have a combination of sinusoidal curves as the family $i - 4$, but with a different frequency f, e.g., with $|F| = 6$ we obtain $F_0 : (X,Y), f = 1$; $F_1 : (Y,X), f = 1$; $F_2 : (X,X), f = 1$; $F_3 : (Y,Y), f = 1.$; $F_4 : (X,Y), f = 2.$; and $F_5 : (Y,X), f = 2$ all having a perturbation $\sigma = 0.1$.

For each dataset, we run the clustering algorithms setting the parameter $k = |F|$ as the number of families. Figure 4a reports the trend of the SSE score. The increasing number of families have only a minor impact on *K-MDTSC* performance, while *k-Shape* cannot correctly group elements of the same family together when $|F| > 5$. Looking at the Adjusted Rand index (Figure 4b), *k-Shape* demonstrates the creation of heterogeneous clusters composed of more than one family. Instead, in most of the cases, *K-MDTSC* clusters the data correctly.



(**a**) SSE score (**b**) Adjusted Rand Index

Figure 4. Number of families $|F|$ stability.

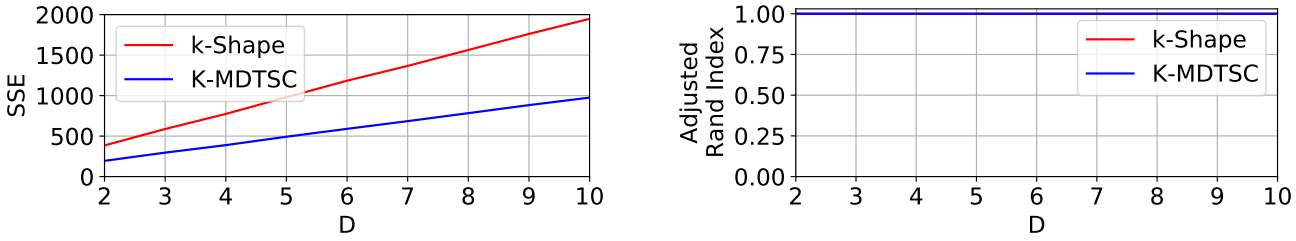
Dimensions Stability: Finally, we study the stability while increasing the number of dimensions D. To this end, we generate datasets composed of 4 families of D-dimensional time-series. We generate datasets with $D\ in [2,10]$, where we characterize each family by a unique combination of sinusoidal curves and sinusoidal periods. As the initial dataset, we keep a $\sigma = 0.1$.

$$Datasets = \begin{cases} F_0 : \begin{cases} (X,Y), f = 1, \sigma = 0.1 \\ (X_i,Y_j), i \in \{2,4,6,8\}, j \in \{3,5,7,9\}, f = Rnd, Rnd \in [2,10], \sigma = 0.1 \end{cases} \\ F_1 : \begin{cases} (Y,X), f = 1, \sigma = 0.1 \\ (Y_i,X_j), i \in \{2,4,6,8\}, j \in \{3,5,7,9\}, f = Rnd, Rnd \in [2,10], \sigma = 0.1 \end{cases} \\ F_2 : \begin{cases} (X,X), f = 1, \sigma = 0.1 \\ (X_i,X_j), i \in \{2,4,6,8\}, j \in \{3,5,7,9\}, f = Rnd, Rnd \in [2,10], \sigma = 0.1 \end{cases} \\ F_3 : \begin{cases} (Y,Y), f = 1, \sigma = 0.1 \\ (Y_i,Y_j), i \in \{2,4,6,8\}, j \in \{3,5,7,9\}, f = Rnd, Rnd \in [2,10], \sigma = 0.1 \end{cases} \end{cases}$$

When $D = 2$, for each family, we keep the same sinusoidal curves as in Equation (1). When $D > 2$, for each family, we repeat the same sinusoidal schema as the first two dimensions but setting the frequency f as a random variable in $[2,10]$. This allows us to create dimensions having different profiles, e.g., when $D = 3$, we define family $F_0 : (X,Y), f = 1, X, f = Rnd, Rnd \in [2,10], \sigma = 0.1$. Hence, as first and second dimension X and Y curves having unitary frequency plus random noise perturbation, and a third dimension being a X curve having a random frequency between 2 and 10 plus a random noise perturbation.

We run *K-MDTSC* and *k-Shape* fixing $k = 4$, equal to the number of families. Both algorithms show that once the clustering algorithm identifies families based on a few di-

mensions, increasing the dimensions does not impact the performance. Indeed, as expected, the SSE score increases (Figure 5a) with more dimensions; however, the Adjusted Rand Index remains stable (Figure 5b). Still, *K-MDTSC* offers more cohesive clusters than *k-Shape* as testified by the SSE score.



(**a**) SSE score (**b**) Adjusted Rand Index

Figure 5. Number of dimensions *D* stability.

Lessons Learned

With synchronous multi-dimensional time-series, the need for alignment drops. When this happens, with a growing noise perturbation or a growing family number, *K-MDTSC* overcomes *k-Shape* performance. Indeed, *K-MDTSC* creates either more cohesive clusters that *k-Shape*, either more homogeneous clusters composed by single multi-dimensional time-series family. Instead, the number of dimensions does not play a significant role, with both algorithms showing perfect family re-identification.

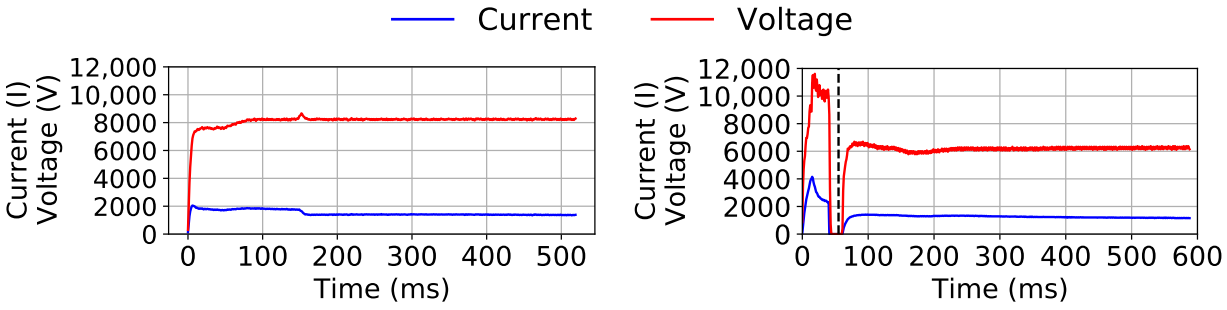
5. Real Use Case

After evaluating *K-MDTSC* with a synthetic dataset, we detail the data collection process in our industrial case study. Then, we define our problem and outline the proposed generic pipeline to preprocess multi-dimensional time-series data.

5.1. Data Collection

To collect data about the manufacturing process, we instrument two robots in two different working stations in the body-in-white shop of a car manufacturing production line. Each robot is composed of one or two *clamps*, each one equipped with a welder soldering different *welding spots* (*ws*) of the car body.

The robot moves through welding spots in a periodic fashion car body after car body. While welding each welding spot *ws*, for each *welding i*, we record the *voltage* $V_{(ws,i)}(z)$, and the *current* $I_{(ws,i)}(z)$ curves as time-series. Time-series samples t are collected with a frequency of 1 kHz. Figure 6 reports two examples of the voltage and current time-series in two different *ws*. While the same welding spot *ws* always follows the same welding process, with possibly little differences only in duration, different welding spots are characterized by different welding processes having different shapes and lengths. Based on $V_{(ws)}(z)$ and $I_{(ws)}(z)$ we define up to two welding phases: *warm-up*, in which the electrode heat up; and the *steady-state* when the welder solder the welding spot. Figure 6a,b reports an example of a one-phase and a two-phase welding process (divided by the vertical dashed line), respectively.



(**a**) One-phase welding process. (**b**) two-phase welding process.

Figure 6. Voltage and current time-series.

While welding, the electrode is continuously exposed to pressure and high temperature, altering the *tip* geometry. Furthermore, during the welding process, material expulsion may soil the electrode, making it less efficient and worsting the welding performance. Therefore, to reshape the electrode and optimize the welding process, two maintenance operations are performed regularly: a *tip dressing* operation and an *electrode replacement*. Firstly, every a fixed number of welds, a *tip dressing* operation is performed. During a tip dressing operation, part of the electrode material is removed to clean it and restore it to its original shape. For each robot clamp, the tip dressing operation is always performed between the same two welding spots. We refer to these two as the *welding spot before* ws_b and the *welding spot after* ws_a the maintenance operation. After each tip dressing operation, a *calibration program* cp runs to calibrate the welder. Each robot runs a different calibration program. The calibration program consists of running a welding operation without any material. We record also the time-series of $V_{(cp,i)}(z)$ and $I_{(cp,i)}(z)$. Secondly, while the tip dressing operation is fundamental to reset the electrode shape, it also reduces the electrode size. Hence, after a fixed number of tip dressing operations, the electrode is replaced with a new one. For each welding operation and each calibration program, we record the relative *wear* index (w). The *wear* index reports the percentage of tip dressing operations performed since the last electrode replacement with respect to the expected number of tip dressing operations per electrode. In a nutshell, the *wear* represents the age of the electrode. We use this index to track maintenance operations. Table 1 summarizes the acquired features.

Table 1. Monitored Parameters.

Parameter	Description
i	When the weld was performed
ws	The welding spot
$I_{(ws,i)}$	Time-series describing the current curve
$V_{(ws,i)}$	Time-series describing the voltage curve
w	Percentage of tip dressing operations with respect to the expected number of tip dressing operation per electrode

We collected data from November 2019 to February 2020 from a real FCA pant. In this period, we record more than 60 thousand welding operations from the instrumented robots. Table 2 summarizes the main dataset information.

Table 2. Dataset Overview.

Working Station	# Clamps	# Welds	Calibration Program ID
Working Station 1	1	22,746	3
Working Station 2	1	26,546	4
Working Station 2	2	89,857	4

5.2. Problem Definition

Our goal is to study the possibility of using this welding data to migrate from periodic maintenance to a predictive maintenance approach. To this end, we address the following research questions:

1. Can we use the welding data to predict the electrode aging and adopt the predictive maintenance?
2. Can we infer further information about the welding process to identify possible patterns hinting at problems?

To answer the questions above, we develop the generic pipeline depicted in Figure 7. First, starting from the data of a selected welding point wp, we design different preprocessing steps to feed *K-MDTSC* and identify clusters of welding operations. Finally, we correlate the clusters with external indexes, i.e., the wear, to identify whether different clusters describe different electrode aging.

In detail, we perform the following steps:

- **Noise reduction:** While *K-MDTSC* can handle any multi-dimensional time-series, domain experts suggested to us to filter the time-series to remove high-frequency noise that depends mostly on the sensors and collection process.
- **Data selection:** Secondly, weldings may have different durations even within the same welding spot. Moreover, two-phase welding spots are characterized by two distinct periods. Hence, we apply different data selection strategies to extract multi-dimensional time-series having (i) the same duration and (ii) that appear synchronized.
- **Data Normalization:** Since $V_{(ws,i)}(z)$ and $I_{(ws,i)}(z)$ have different operational ranges, we run a classic data normalization process based on z-score.
- **Time-Series Clustering:** We run *K-MDTSC* to group welding samples and identify different welding shapes within each welding spot.
- **Correlation with external indexes:** Finally, we study the clusters by using the wear to discover whether different welding shapes are characterized by different tears and wear of the electrode.

Figure 7. Pipeline of the data analysis.

In the following we detail each step.

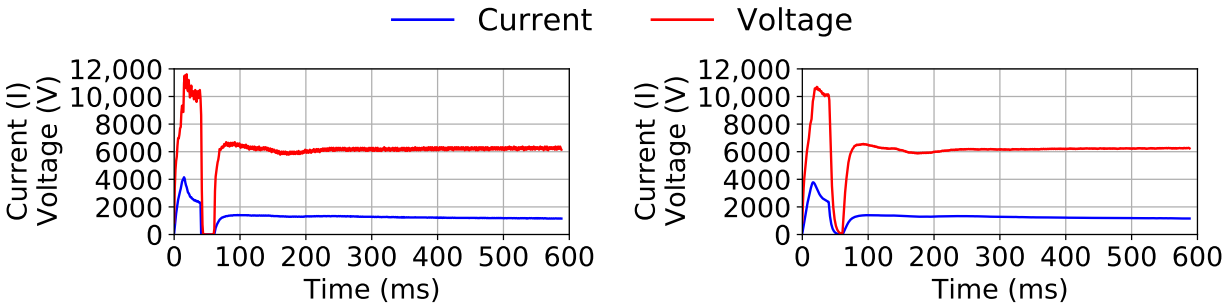
5.3. Noise Reduction

In the data collection the voltage $V_{(ws,i)}(z)$ and current $I_{(ws,i)}(z)$ time-series are collected with a frequency of 1 kHz. Such a moderately high frequency allows us to spot sudden and short changes. However, in the presence of noise in the acquisition system, the samples may suffer from frequent changes although the welder's output remains stable. As such, to remove the noise component, we apply a low pass filter. In detail, given a

time-series $X_{(ws,i)}$ with $X \in \{V, I\}$, we relay on a exponential moving average function defined as follows:

$$\bar{X}_{(ws,i)}(z+1) = \alpha * X_{(ws,i)}(z) + (1-\alpha) * \bar{X}_{(ws,i)}(z)$$

where $\bar{X}_{(ws,i)}(z)$ is the value of the exponential moving average at time z, α is the smoothing factor. Notice that the higher α, the more weight is given to the recent observation, i.e., the noise component, while the lower the alpha, the more we average the signal. Figure 8 reports an example of the noise reduction effect with $\alpha = 0.75$.



(**a**) Before (**b**) After

Figure 8. Noise reduction.

5.4. Data Selection

Given a welding point wp, the different welding samples i may be described by curves $\bar{X}_{(ws,i)}(z)$ with $\bar{X} \in \{\bar{V}, \bar{I}\}$ having different durations. Furthermore, for welding points wp characterized by a two-phases welding process, the *steady-state* may begin in different instant z_0 based on the *warm-up* duration. Since our clustering solution requires that all the time-series in the input have the same duration and are synchronized, we align all the time-series as follows:

Whole Time-Series: Firstly, we consider the entire time-series. Given a welding point wp, we select the entire time-series $\bar{X}_{(ws,*)}(z)$. Next, we select the earliest finish time-series $E_f(ws)$ as the earliest time for a welding sample to finish.

$$E_f(ws) = min(|\bar{X}_{(ws,*)}|)$$

Finally, for each time-series $i \in \bar{X}_{(ws,*)}$, we get $\hat{X}_{(ws,i)}(z)$ as the time-series lasting $E_f(ws)$.

$$\hat{X}_{(ws,i)}(z) = \bar{X}_{(ws,i)}(z) \;\; 0 \leq z \leq E_f(ws)$$

Figure 9a depicts the results of the data selection step for all the time-series of the welding spot shown in Figure 8b.

Steady-State Aligned: Secondly, we focus only on the *steady-state* as the actual phase when the welder solder. Given a welding spot wp, for each time-series $\bar{X}_{(ws,i)}(z)$ we identify the *start of the steady-state* $S_{st}(ws,i)$ as the time when the welding process starts after the warm-up phase ends. We identify $S_{st}(ws,i)$ based on the current curve $\bar{I}_{(ws,i)}(z)$. Given $\bar{I}_{(ws,i)}(z)$, the warm-up phase ends when the current value drops below a given $Threshold_I$. Next, we identify $S_{st}(ws,i)$ as the time when the current value $\bar{I}_{(ws,i)}(z)$ overcomes the $Threshold_I$ again.

For each time-series $i \in \bar{X}_{(ws,*)}$, we get the steady-state $\tilde{X}_{(ws,i)}$ as follow:

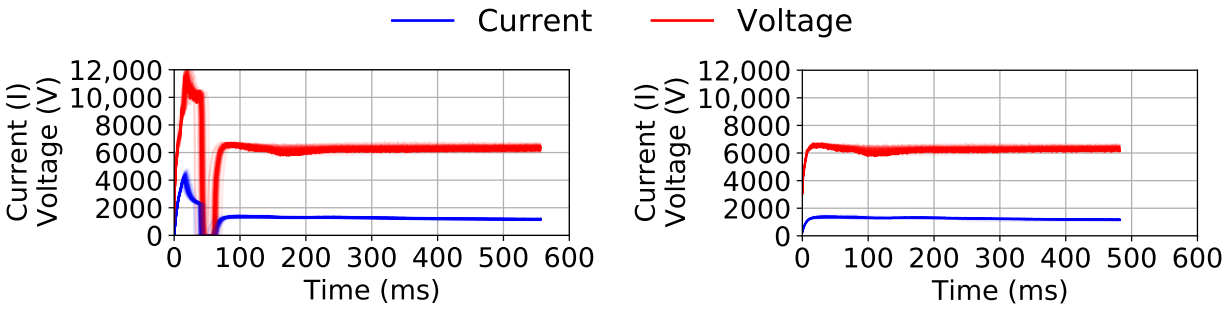
$$\tilde{X}_{(ws,i)}(z) = \bar{X}_{(ws,i)}(z) \;\; z \geq S_{st}(ws,i)$$

After this, the *steady-states* are aligned, i.e., all $\tilde{X}_{(ws,i)}(z)$ start synchronously. However, since they may have different duration, we compute the earliest finish $\tilde{E_f}(ws)$ over all $\tilde{X}_{(ws,i)}(z)$.

Finally, for each time-series $i \in \tilde{X}_{(ws,*)}$, we get the *steady-state* $\tilde{X}_{(ws,i)}(z)$ lasting $E_f(ws)$.

$$\tilde{X}_{(ws,i)}(z) = \tilde{X}_{(ws,i)}(z) \;\; 0 \leq z \leq \tilde{E_f}(ws)$$

Figure 9a depicts the result of the *steady-states* data selection step for the time-series shown in Figure 8b.



(a) $\hat{X}_{(ws,i)}(z)$ **(b)** Steady-State aligned

Figure 9. Data Selection.

5.5. Data Normalization

After the data selection, the time-series $\hat{V}_{(ws,i)}(z)$ and $\hat{I}_{(ws,i)}(z)$ (or $\tilde{V}_{(ws,i)}(z)$ and $\tilde{I}_{(ws,i)}(z)$) are represented with different ranges. As such, we run a standard z-score normalization process as follows.

Given a welding sample ws, we select all the time-series $\hat{V}_{(ws,i)}(z)$ and $\hat{I}_{(ws,i)}(z)$. We create two time-series $\hat{V}_{ws}(z)$ and $\hat{I}_{ws}(z)$, where each one is the union of all $\hat{V}_{(ws,*)}(z)$ and $\hat{I}_{(ws,*)}(z)$ selected in the temporal order, respectively.

$$\hat{V}_{ws}(z) = \bigcup \hat{V}_{(ws,i)}(z) \;\; i \in (1, .., |\hat{V}_{ws}|)$$

$$\hat{I}_{ws}(z) = \bigcup \hat{I}_{(ws,i)}(z) \;\; i \in (1, .., |\hat{I}_{ws}|)$$

Next, from $\hat{I}_{ws}(z)$ and $\hat{V}_{ws}(z)$, we extract the:

- *Mean*: $m = \frac{1}{N} \sum_{z \in w(z)} x_i(z)$;
- *Standard deviation*: $\sigma = \sqrt{\frac{1}{N-1} \sum_{z \in w(z)} (x_i(z) - m)^2}$

Finally, we run the z-score normalization separately for each welding sample i:

$$\hat{V}_{ws,i}(z) = \frac{\hat{V}_{(ws,i)}(z) - m_V}{\sigma_V} \;\; i \in (1, .., |\hat{V}_{ws}|)$$

$$\hat{I}_{ws,i}(z) = \frac{\hat{I}_{(ws,i)}(z) - m_I}{\sigma_I} \;\; i \in (1, .., |\hat{I}_{ws}|)$$

6. Case Study Evaluation

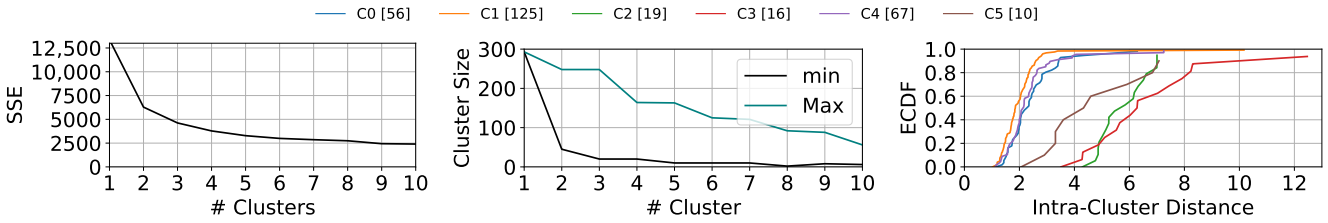
We evaluate the proposed pipeline on the real data collected from a real production line as described in Section 5.

6.1. First Robot

Firstly, we focus on the data of the robot in Working Station 1 composed of a single clamp. Since the maintenance operation's highest impact should be more evident in the welding point immediately before (wp_b) or after (wp_a) the maintenance operation, we focus

on them. Considering the welding technique, both use a one-phase welding process, hence after the noise reduction step, we use the *Whole Time-Series* data selection process.

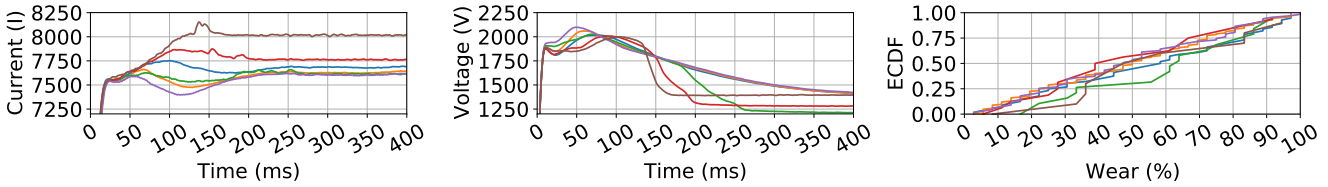
Focusing on wp_b, Figure 10a depicts the trend of the SSE score, while Figure 10b reports the trend of the minimum and maximum clusters size while increasing the number of clusters k. We use these two metrics to choose the best k. The SSE score suggests that increasing k by more than 4 gives a marginal benefit as the SSE keeps reducing at a slower pace while increasing k. However, considering the difference between the smallest and the biggest clusters, we can see that with more than 5 clusters, we start having a stable minimum cluster size. In contrast, the maximum cluster size starts reducing at a slower pace. Hence, to study the clustering result, we set $k = 6$.



(**a**) SSE (**b**) Cluster Size (**c**) Intra-Cluster Distance

Figure 10. Cluster Metrics Working Station 1 Clamp 1 ws_b.

Next, we focus on the intra-cluster distance as a quality metric of the clustering result. We focus on this metric as huge distances may highlight unusual welding shapes, thus requiring maintenance operation. Figure 10c reports the Empirical Cumulative Distribution Function (ECDF) on the $y - axis$ of the intra-cluster distance separately for each cluster. The legend also reports the size of each cluster. Clusters $C0$, $C1$, and $C4$, accounting in total for 248 out of 293 welding samples, show very dense results with limited intra-cluster distance. On the contrary, small clusters show the highest distance. To investigate these results, we focus on each cluster's centroid separately for the current and the voltage curves. Figure 11a,b show the denormalized current and the voltage centroids, respectively. Cluster $C5$ clearly shows higher values of the current with respect to the other clusters. Clusters $C2$ and $C3$, instead, show a higher deviation from the other clusters by looking at the voltage curves.



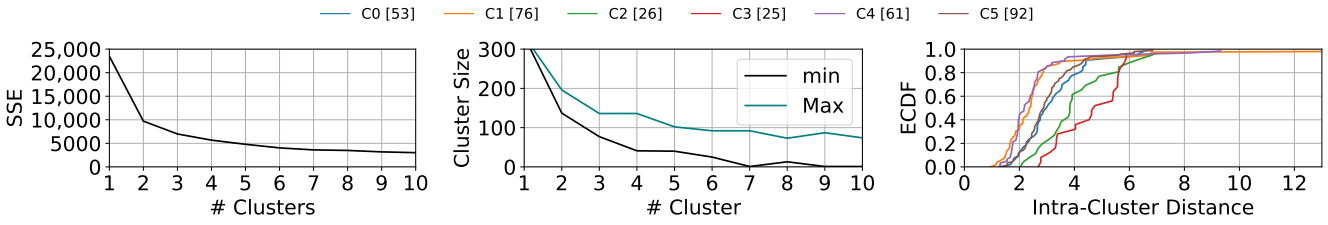
(**a**) Current Centroids (**b**) Voltage Centroids (**c**) Correlation with wear

Figure 11. Clusters representation Working Station 1 Clamp 1 ws_b.

To identify whether these differences also reflect on the wear, Figure 11c reports, for each cluster, the ECDF of the wear of the welding samples. In a nutshell, we evaluate whether different clusters are characterized by different wear of the electrode. Thus, if we can use the clustering result to identify welding shapes that clearly point to maintenance needs, in this case, we could trigger the maintenance operation only when such welding shapes appear. Unlike the clustering results, however, the correlation with the wear does not show a clear picture. Clusters tend to be heterogeneous concerning this metric, with welding samples having different wear and tear of the electrode present in all the clusters.

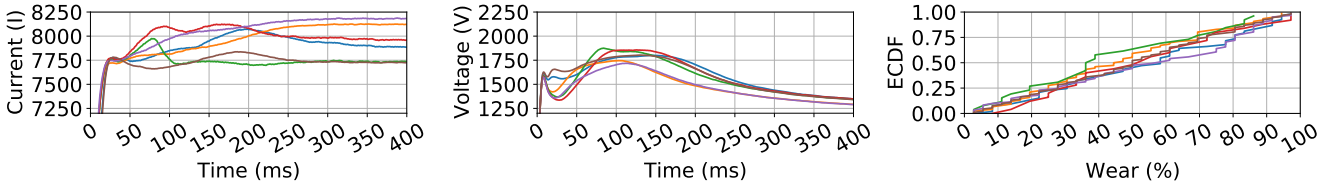
Next, we focus on wp_a. The SSE score (Figure 12a) reports a trend similar to the previous case. Hence, to identify the best value of k we use the trend of the clusters size

(Figure 12b). After $k = 5$, the maximum size trend starts stabilizing while the minimum size drops after $k = 6$. Thus, we set $k = 6$. Differently from the previous case, only a few time-series in clusters have a large distance, as shown by the tail of the distribution in Figure 12c. This suggests a good clustering result with centroids well repressing the respective clusters. Here, cluster $C5$ is the most heterogeneous one, with 25 time-series. Considering the current centroids in Figure 13a, they clearly show differences among them, while voltage centroids in Figure 13b are more similar. By studying the correlation with the wear, in Figure 13c, all clusters contain heterogeneous welding samples. This hints that both the welding points before and after the maintenance operations do not show any particular impact on the electrode's wear and tear.



(**a**) SSE (**b**) Cluster Size (**c**) Intra-Cluster Distance

Figure 12. Cluster Metrics Working Station 1 Clamp 1 ws_a.



(**a**) Current Centroids (**b**) Voltage Centroids (**c**) Correlation with wear

Figure 13. Clusters representation Working Station 1 Clamp 1 ws_a.

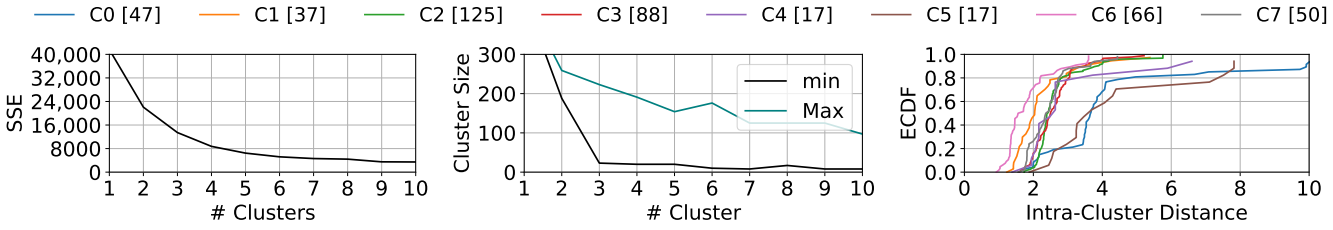
Given these results, we also study calibration program 3 of the robot. The results, not shown here for the sake of brevity, highlight that the calibration program is not a good proxy of the maintenance operation. By discussing with domain experts, this may be because the calibration program runs without any material, with limited electrode usage.

6.2. Second Robot

We now repeat the analysis to study whether our results are generalized to the second robot. We focus on a welding point of the robot in the Working Station 2, Clamp 1. We focus on a welding point characterized by a two-phase welding process. We start by employing the *Whole Time-Series* data selection. Figure 14a reports the trend of the SSE score. The presence of the *warm-up* phase is immediately visible. The SSE score reaches a maximum value almost twice as much as the previous welding spot. The shape of the SSE score suggests that we should consider at least 4 clusters. Intuitively, Figure 14b shows that the higher k, the smaller is the maximum cluster size. From this metric, we choose $k = 8$ as the maximum cluster size presents a plateau while the minimum cluster size presents a maximum local value. By choosing $k = 8$, clusters $C0$ and $C5$ present, generally, a higher intra-cluster distance (Figure 14c). Analyzing the clusters' centroid, we can see that this is due mostly to different peaks in the voltage curve (Figure 15a) rather than in the current curve (Figure 14c). Similarly, with the ws_b case in Figure 11c, also in Figure 15c, the wear does not reflect the intra-cluster distance behavior. Indeed, cluster $C0$ presents an ECDF of the wear as the other clusters.

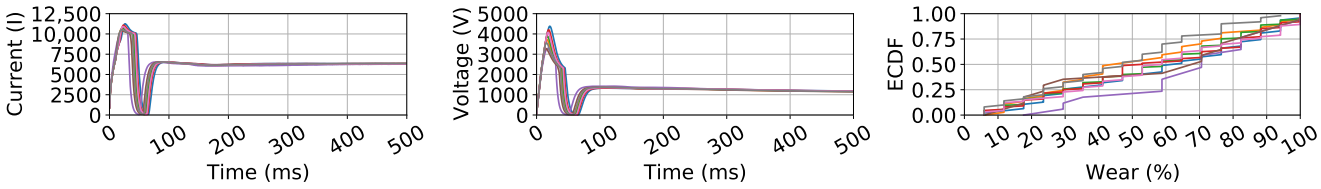
Next, since the *warm-up* phase demonstrates to drive the clustering, we rely on the *Steady-State aligned* data selection to focus on the actual welding process. We analyze the

same two-phase welding spot to study the differences among the two clustering results. Figure 16a reports the trend of the SSE score. Clearly, the SSE score reaches a lower value than the previous case, slowing down in the reduction pace with at least 4 clusters. Looking at Figure 16b, we set $k = 6$, as the minimum cluster size presents a plateau while the maximum cluster size keeps decreasing. Interestingly, the intra-cluster distance in Figure 16c shows that cluster C0 groups time-series that are always very far from their centroid. However, the current centroid in Figure 17a shows a shape very similar to the other clusters. Differently, the voltage centroid in Figure 17b, shows a very different shape with a lower voltage value during the actual welding process. As in the other cases, the ECDF of the wear, Figure 17c does not show any particular behavior related to the different welding shapes.



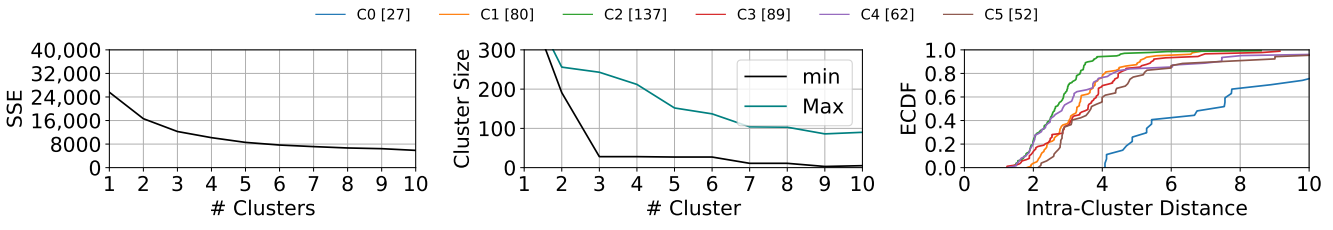
(**a**) SSE (**b**) Cluster Size (**c**) Intra-Cluster Distance

Figure 14. Cluster Metrics *wp* Working Station 2 Clamp 1, Whole Time-Series Data Selection.



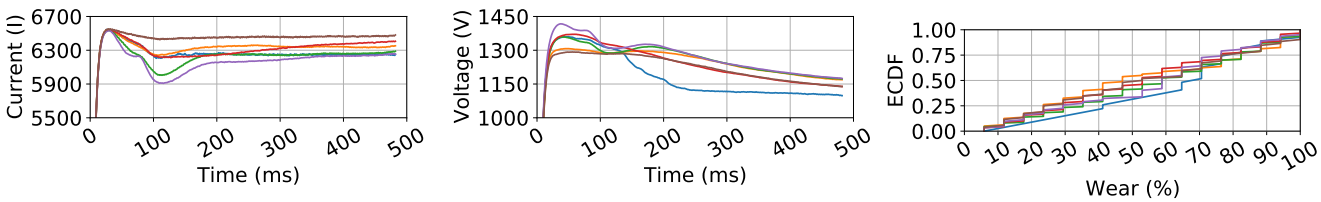
(**a**) Current Centroids (**b**) Voltage Centroids (**c**) Correlation with wear

Figure 15. Clusters representation *wp* Working Station 2 Clamp 1, Whole Time-Series Data Selection.



(**a**) SSE (**b**) Cluster Size (**c**) Intra-Cluster Distance

Figure 16. Cluster Metrics *wp* Working Station 2 Clamp 1, Steady-State aligned Data Selection.



(**a**) Current Centroids (**b**) Voltage Centroids (**c**) Correlation with wear

Figure 17. Clusters representation *wp* Working Station 2 Clamp 1, Steady-State aligned Data Selection.

By analyzing the results with domain experts, we discover that current scheduled maintenance is performed with a frequency higher than required. Our results, therefore, confirm that this behavior allows the robots to weld without altering the welding shape while increasing the wear and tear of the electrode. As a result, domain experts plan to increase the time gap between two maintenance operations. This increase will result in a time-reduction of the time spent for the maintenance operations and money-saving due to increased electrodes' usage.

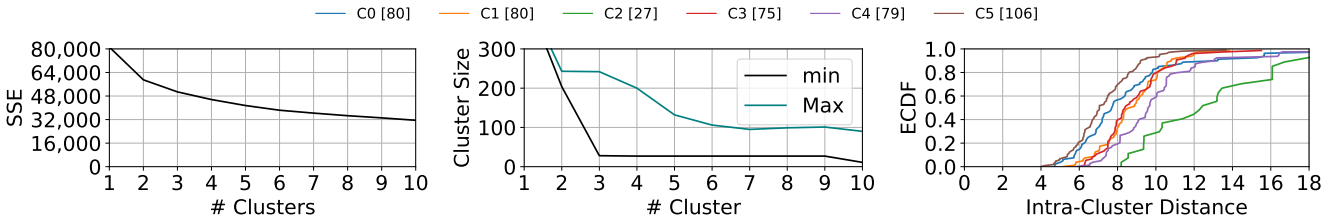
Lessons Learned

Our automatic pipeline effectively preprocesses and correctly identifies different welding profiles. Different welding profiles are not characterized by homogeneous wear of the electrode. This heterogeneity hints that the manufacturer could postpone current maintenance until the wear does not clearly impact the welding shape. This result is generalized over the different welding points and robots. Furthermore, with the two-phase welding process, the *steady-state* demonstrates to hold most of the information. Thus, when possible, it is better to study the welding shape only on this part of the welding process.

6.3. Noise Sensitivity

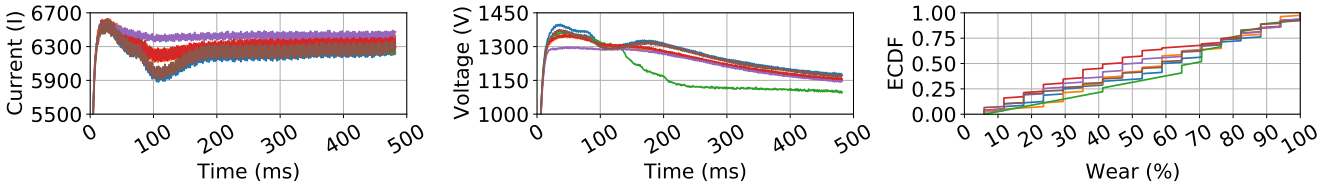
Since *K-MDTSC* could handle noisy curves, here we study the impact of the noise reduction step introduced in our case study. To this end, we repeat the clustering process of the welding point collected by the robot in Working Station 2. We repeat our generic pipeline, avoiding the noise reduction step and applying the *steady-state aligned* data selection process. Comparing the SSE score in Figure 16a (with noise reduction) and Figure 18a (without noise reduction), we can see that the curves have a similar trend. However, when we do not apply the noise reduction, the maximum value increases almost three times. The cluster size (Figure 18b), instead, shows both similar trends and clusters cardinality with respect to the case with noise reduction (Figure 16c). Hence, we set $k = 6$. The clustering result shows that clusters are much less dense than the case with noise reduction. Indeed, in Figure 18c, we can see how all intra-cluster distances are greater than 4. Instead, in Figure 16c, most of the intra-cluster distances are lower than 4. Considering the cluster centroids, in Figure 19a,b, the identification of different welding shapes is a complex task, as the noise gives a major contribution, especially in the current curve. Finally, as in the previous cases, also here (Figure 19c), different welding shapes do not reflect differences in the electrode's wear.

Based on these results, on the one hand, *K-MDTSC* confirms the capability to handle noise data. On the other hand, a noise reduction step demonstrates to ease the analysis of the clustering results.



(**a**) SSE (**b**) Cluster Size (**c**) Intra-Cluster Distance

Figure 18. Cluster Metrics *wp* Working Station 2 Clamp 1, no noise reduction.



(**a**) Current Centroids (**b**) Voltage Centroids (**c**) Correlation with wear

Figure 19. Clusters representation *wp* Working Station 2 Clamp 1, no noise reduction .

Lessons Learned

The noise reduction step demonstrates limited benefits, mainly visible while considering the cluster cohesiveness and the centroids representation. Instead, the correlation with the electrode wear does not show any impact of the noise. Indeed, also in this case the electrode wear does not alter the welding shape.

7. Conclusions

This work presented *K-MDTSC*, a K-means-based clustering algorithm designed to group synchronous multi-dimensional time-series. Firstly, we ran a thorough validation process using synthetic datasets composed of synchronous multi-dimensional sinusoidal curves and compared the results with *k-Shape*, a state-of-art time-series clustering algorithm K-means-based. Our results showed that:

- both *K-MDTSC* and *k-Shape* correctly group multi-dimensional time-series;
- however, *K-MDTSC* overcomes *k-Shape* performance when a high number of curve families or noise perturbation are present.

Next, we used *K-MDTSC* to analyze data from the body-in-white welding stage of an actual production plant. To use *K-MDTSC* with real data, we deploy a generalized pipeline that reduces the noise in the multi-dimensional time-series, realign them if needed to get synchronous multi-dimensional time-series, normalize each dimension separately, and finally fed them into *K-MDTSC*. Our results show that our automatic pipeline effectively processes the data and that *K-MDTSC* algorithm identifies different welding profiles. By presenting our results to domain experts, they discover that:

- the wear and tear of the electrode do not negatively impact the welding process;
- considering the different data selection techniques, the wear and tear of the electrode does not negatively impact it neither considering the whole welding process nor considering only the steady-state when the actual welding process is performed;

Based on these results, experts reach the conclusion that current maintenance operations are premature and could be postponed with a reduction of the time spent for maintenance and a resulting money-saving.

To allow other researchers to replicate our results and use *K-MDTSC* to run similar analyses with synchronous multi-dimensional time-series, we release the *K-MDTSC* code and our synthetic datasets as open-source at [8].

As future work, we plan to extend our validation process to other public multi-dimensional time-series datasets and run a thoroughly compare *K-MDTSC* algorithm with other time-series clustering algorithms. For future analysis, we plan to extend our analyses to the new data characterized by postponed maintenance operations.

Author Contributions: Conceptualization, M.M., T.C. and D.G.; investigation, D.G., M.M. and T.C.; methodology, M.M. and D.G.; software, D.G.; supervision, M.M and T.C.; writing—original draft, D.G., T.C. and M.M.; writing—review and editing, M.M., T.C. and D.G. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results is supported by the SmartData@PoliTO center for data analysis and Big Data technologies and by the Centro di Ricerche Fiat (CRF) through a research project.

Data Availability Statement: Data to reproduce synthetic experiments are available at https://github.com/smartdatapolito/K-MDTSC, accessed on 1 May 2021.

Acknowledgments: We thank Luca Tomaino for the preliminary analysis performed for the case study application and Giorgio Giorgio Pasquettaz for his help in case study application and the project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hill, T.; O'Connor, M.; Remus, W. Neural network models for time series forecasts. *Manag. Sci.* **1996**, *42*, 1082–1092.
2. Bhandari, S.; Bergmann, N.; Jurdak, R.; Kusy, B. Time series data analysis of wireless sensor network measurements of temperature. *Sensors* **2017**, *17*, 1221.
3. Wei, L.; Keogh, E. Semi-supervised time series classification. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006.
4. Gopalapillai, R.; Gupta, D.; Sudarshan, T. Experimentation and analysis of time series data for rescue robotics. In *Recent Advances in Intelligent Informatics*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 443–453.
5. Aghabozorgi, S.; Seyed Shirkhorshidi, A.; Ying Wah, T. Time-series clustering—A decade review. *Inf. Syst.* **2015**, *53*, 16–38.
6. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*; Statistical Laboratory of the University of California: Berkeley, CA, USA, 1967.
7. Paparrizos, J.; Gravano, L. k-Shape: Efficient and Accurate Clustering of Time Series. *ACM Sigmod Rec.* **2016**, *45*, 69–76.
8. Smartdatapolito. K-MDTSC: K-Multi-Dimensional Time-Series Clustering Algorithm. 2021. Available online: https://github.com/smartdatapolito/K-MDTSC (accessed on 7 April 2021).
9. Celenk, M. A color clustering technique for image segmentation. *Comput. Vis. Graph. Image Process.* **1990**, *52*, 145–170.
10. Chuang, K.S.; Tzeng, H.L.; Chen, S.; Wu, J.; Chen, T.J. Fuzzy c-means clustering with spatial information for image segmentation. *Comput. Med. Imaging Graph.* **2006**, *30*, 9–15.
11. Dhanachandra, N.; Manglem, K.; Chanu, Y.J. Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. *Procedia Comput. Sci.* **2015**, *54*, 764–771.
12. Glowacz, A. Ventilation Diagnosis of Angle Grinder Using Thermal Imaging. *Sensors* **2021**, *21*, 2853, doi:10.3390/s21082853.
13. Huang, A. Similarity measures for text document clustering. In Proceedings of the Sixth New Zealand Computer Science Research Student Conference, Christchurch, New Zealand, 14 April 2008.
14. Beil, F.; Ester, M.; Xu, X. Frequent term-based text clustering. In Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002.
15. Faroughi, A.; Javidan, R.; Mellia, M.; Morichetta, A.; Soro, F.; Trevisan, M. Achieving horizontal scalability in density-based clustering for URLs. In Proceedings of the IEEE International Conference on Big Data, Seattle, WA, USA, 10–13 December 2018.
16. Giordano, D.; Traverso, S.; Grimaudo, L.; Mellia, M.; Baralis, E.; Tongaonkar, A.; Saha, S. YouLighter: A cognitive approach to unveil YouTube CDN and changes. *IEEE Trans. Cogn. Commun. Netw.* **2015**, *1*, 161–174.
17. Morichetta, A.; Mellia, M. Clustering and evolutionary approach for longitudinal web traffic analysis. *Perform. Eval.* **2019**, *135*, 102033.
18. Giordano, D.; Traverso, S.; Grimaudo, L.; Mellia, M.; Baralis, E.; Tongaonkar, A.; Saha, S. Youlighter: An unsupervised methodology to unveil youtube cdn changes. In Proceedings of the 2015 27th International Teletraffic Congress, Ghent, Belgium, 8–10 Septemebr 2015; pp. 19–27.
19. Chen, H.; Yin, H.; Li, X.; Wang, M.; Chen, W.; Chen, T. People opinion topic model: Opinion based user clustering in social networks. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017.
20. Li, P.; Dau, H.; Puleo, G.; Milenkovic, O. Motif clustering and overlapping clustering for social network analysis. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017.
21. Curiskis, S.A.; Drake, B.; Osborn, T.R.; Kennedy, P.J. An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit. *Inf. Process. Manag.* **2020**, *57*, 102034.
22. Huang, L.; Shea, A.L.; Qian, H.; Masurkar, A.; Deng, H.; Liu, D. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *J. Biomed. Inform.* **2019**, *99*, 103291.
23. Yelipe, U.; Porika, S.; Golla, M. An efficient approach for imputation and classification of medical data values using class-based clustering of medical records. *Comput. Electr. Eng.* **2018**, *66*, 487–504.
24. Sun, W.; Cai, Z.; Liu, F.; Fang, S.; Wang, G. A survey of data mining technology on electronic medical records. In Proceedings of the IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, 12–15 October 2017.
25. Hautamaki, V.; Nykanen, P.; Franti, P. Time-series clustering by approximate prototypes. In Proceedings of the 19th International Conference on Pattern Recognition, Tampa, FL, USA, 8–11 December 2008.
26. Ghassempour, S.; Girosi, F.; Maeder, A. Clustering multivariate time series using hidden Markov models. *Int. J. Environ. Res. Public Health* **2014**, *11*, 2741–2763.

27. Zakaria, J.; Mueen, A.; Keogh, E. Clustering time series using unsupervised-shapelets. In Proceedings of the IEEE 12th International Conference on Data Mining, Brussels, Belgium, 10–13 December 2012.
28. Rakthanmanon, T.; Keogh, E.J.; Lonardi, S.; Evans, S. MDL-based time series clustering. *Knowl. Inf. Syst.* **2012**, *33*, 371–399.
29. Ding, H.; Trajcevski, G.; Scheuermann, P.; Wang, X.; Keogh, E. Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. *Proc. Vldb Endow.* **2008**. *1*, 1542–1552.
30. Vlachos, M.; Lin, J.; Keogh, E.; Gunopulos, D. A wavelet-based anytime algorithm for k-means clustering of time series. In Proceedings of the Workshop on Clustering High Dimensionality Data and Its Applications, San Francisco, CA, USA, 3 May 2003.
31. Wang, X.; Smith, K.A.; Hyndman, R.J. Dimension reduction for clustering time series using global characteristics. In Proceedings of the International Conference on Computational Science, Atlanta, GA, USA, 22–25 May 2005.
32. Abonyi, J.; Feil, B.; Nemeth, S.; Arva, P. Principal component analysis based time series segmentation. In Proceedings of the IEEE International Conference on Computational Cybernetics, Hotel Le Victoria, Mauritius, 13–16 April 2005.
33. Fu, T.c.; Chung, F.l.; Ng, V.; Luk, R. Pattern discovery from stock time series using self-organizing maps. In Proceedings of the Workshop Notes of KDD2001 Workshop on Temporal Data Mining, San francisco, CA, USA, 26–29 August 2001.
34. Kumar, M.; Patel, N.R.; Woo, J. Clustering seasonality patterns in the presence of errors. In Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002.
35. Dias, J.G.; Vermunt, J.K.; Ramos, S. Clustering financial time series: New insights from an extended hidden Markov model. *Eur. J. Oper. Res.* **2015**, *243*, 852–864.
36. Sadahiro, Y.; Kobayashi, T. Exploratory analysis of time series data: Detection of partial similarities, clustering, and visualization. *Comput. Environ. Urban Syst.* **2014**, *45*, 24–33.
37. Ji, M.; Xie, F.; Ping, Y. *A Dynamic Fuzzy Cluster Algorithm for Time Series*; Abstract and Applied Analysis; Hindawi: London, UK, 2013; Volume 2013.
38. Horenko, I. On clustering of non-stationary meteorological time series. *Dyn. Atmos. Ocean.* **2010**, *49*, 164–187.
39. Wismüller, A.; Lange, O.; Dersch, D.R.; Leinsinger, G.L.; Hahn, K.; Pütz, B.; Auer, D. Cluster analysis of biomedical image time-series. *Int. J. Comput. Vis.* **2002**, *46*, 103–128.
40. Möller-Levet, C.S.; Klawonn, F.; Cho, K.H.; Wolkenhauer, O. Fuzzy clustering of short time-series and unevenly distributed sampling points. In Proceedings of the International Symposium on Intelligent Data Analysis, Berlin, Germany, 28–30 August 2003.
41. Liao, T.W. Clustering of time series data—A survey. *Pattern Recognit.* **2005**, *38*, 1857–1874.
42. Javed, A.; Lee, B.S.; Rizzo, D.M. A benchmark study on time series clustering. *Mach. Learn. Appl.* **2020**, *1*, 100001.
43. Bukhsh, Z.A.; Saeed, A.; Stipanovic, I.; Doree, A.G. Predictive maintenance using tree-based classification techniques: A case of railway switches. *Transp. Res. Part C Emerg. Technol.* **2019**, *101*, 35–54.
44. Renga, D.; Apiletti, D.; Giordano, D.; Nisi, M.; Huang, T.; Zhang, Y.; Mellia, M.; Baralis, E. Data-driven exploratory models of an electric distribution network for fault prediction and diagnosis. *Computing* **2020**, *1*, 1–13.
45. Verhagen, W.J.; De Boer, L.W. Predictive maintenance for aircraft components using proportional hazard models. *J. Ind. Inf. Integr.* **2018**, *12*, 23–30.
46. Markudova, D.; Mishra, S.; Cagliero, L.; Vassio, L.; Mellia, M.; Baralis, E.; Salvatori, L.; Loti, R. Preventive maintenance for heterogeneous industrial vehicles with incomplete usage data. *Comput. Ind.* **2021**, *130*, 103468, doi:10.1016/j.compind.2021.103468.
47. Giordano, D.; Pastor, E.; Giobergia, F.; Cerquitelli, T.; Baralis, E.; Mellia, M.; Neri, A.; Tricarico, D. Dissecting a Data-driven Prognostic Pipeline: A Powertrain use case. *Expert Syst. Appl.* **2021**, *180*, 115109.
48. Tessaro, I.; Mariani, V.C.; Coelho, L.d.S. Machine Learning Models Applied to Predictive Maintenance in Automotive Engine Components. *Multidiscip. Digit. Publ. Inst. Proc.* **2020**, *64*, 26.
49. Glowacz, A. Acoustic fault analysis of three commutator motors. *Mech. Syst. Signal Process.* **2019**, *133*, 106226.
50. Panicucci, S.; Nikolakis, N.; Cerquitelli, T.; Ventura, F.; Proto, S.; Macii, E.; Makris, S.; Bowden, D.; Becker, P.; O'Mahony, N.; et al. A Cloud-to-Edge Approach to Support Predictive Analytics in Robotics Industry. *Electronics* **2020**, *9*, 492, doi:10.3390/electronics9030492.
51. Bousdekis, A.; Lepenioti, K.; Apostolou, D.; Mentzas, G. A Review of Data-Driven Decision-Making Methods for Industry 4.0 Maintenance Applications. *Electronics* **2021**, *10*, 828, doi:10.3390/electronics10070828.
52. Uhlmann, E.; Pontes, R.P.; Geisert, C.; Hohwieler, E. Cluster identification of sensor data for predictive maintenance in a Selective Laser Melting machine tool. *Procedia Manuf.* **2018**, *24*, 60–65.
53. Amruthnath, N.; Gupta, T. A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In Proceedings of the 5th International Conference on Industrial Engineering and Applications, Singapore, 26–28 April 2018.
54. Kanawaday, A.; Sane, A. Machine learning for predictive maintenance of industrial machines using IoT sensor data. In Proceedings of the 8th IEEE International Conference on Software Engineering and Service Science, Beijing, China, 24–26 November 2017.
55. Kulkarni, K.; Devi, U.; Sirighee, A.; Hazra, J.; Rao, P. Predictive maintenance for supermarket refrigeration systems using only case temperature data. In Proceedings of the Annual American Control Conference, Milwaukee, WI, USA, 27–29 June 2018.
56. Jimenez-Cortadi, A.; Irigoien, I.; Boto, F.; Sierra, B.; Rodriguez, G. Predictive maintenance on the machining process and machine tool. *Appl. Sci.* **2020**, *10*, 224.

57. Jie, Y.; Qiang, Y. Integrating hidden Markov models and spectral analysis for sensory time series clustering. In Proceedings of the Fifth IEEE International Conference on Data Mining, Houston, TX, USA, 27–30 November 2005.
58. Tan, P.N.; Steinbach, M.; Karpatne, A.; Kumar, V. *Introduction to Data Mining*, 2nd ed.; Pearson: London, UK, 2018.
59. Yeung, K.Y.; Ruzzo, W.L. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics* **2001**, *17*, 763–774.
60. Scikit Learn. Adjusted Rand Score. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html (accessed on 7 April 2021).