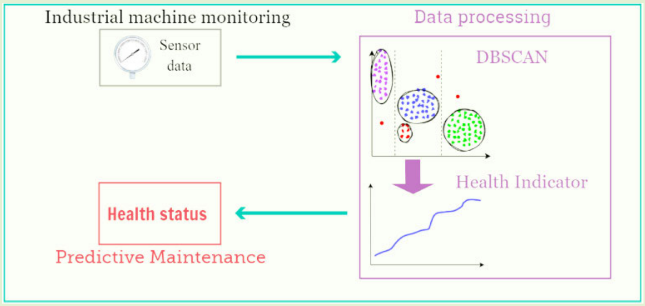
On Sensor Data Clustering for Machine Status Monitoring and Its Application to Predictive Maintenance

Eleonora Oliosi, Gabriele Calzavara, and Gianluigi Ferrari, *Senior Member, IEEE*

***Abstract*—Predictive maintenance is one of the main approaches on which Industry 4.0 is based since it aims at reducing unplanned downtime and maintenance costs of industrial machines. In this work, a time-aware clustering-based approach to the analysis of sensor data is presented for the purpose of monitoring the time evolution of the health status of an industrial machine. A possible application of the proposed framework to predictive maintenance is then proposed. As a relevant representative application scenario, the focus is on one of the key machines in a pharmaceutical plant: a freeze dryer. The illustrated procedure allows for carrying out a time segmentation of the properly sensed data. More precisely, the corresponding operational points (associated with features of the sensed data) are clustered using various algorithms, among which density-based spatial clustering of applications with noise (DBSCAN) turns out to be the best. The benefits of the proposed approach are: 1) its general nature and 2) the limited amount of needed features that have to be extracted from a single sensor signal. The proposed procedure is attractive when the collected data (e.g., from a single sensor) are not sufficient to build an accurate physical model of the monitored component.**



***Index Terms*—Clustering, density-based spatial clustering of applications with noise (DBSCAN), predictive maintenance, sensor data processing.**

I. Introduction

IN INDUSTRIAL pharmaceutical plants, the use of heterogeneous sensors to monitor the production processes is nowadays common. The sensed data are usually recorded for years [1]. The historical process data of an industrial plant can be specifically used to analyze the behavior of the components of the plant itself [2], [3], [4]. The most typical strategies involve training anomaly classifiers and building predictive maintenance algorithms based on collected data [5], [6]. The efficiency of the developed models is heavily affected by the collected data and the type of components to be monitored [7].

Predictive maintenance is a methodology that aims at predicting the deterioration of the health conditions of an industrial machine, typically associated with anomalies in its components. Predicting accurately impending failures can be very difficult: it is essential to have a deep knowledge of the specific system to derive a precise prediction model [8]. Nevertheless, it may happen that the collected data provide inadequate information to accurately determine the degradation status of a particular component. In fact, the recorded sensor data are often representative of the status of a combination of components. Accurate knowledge of system physics may be necessary to detect the origin of a variation in an inspected sensor signal [9]. Therefore, a variation of the operational condition of the machine can easily be detected, but the difficulty lies in the identification of the specific responsible machine component.

This work represents a significant extension of [10], where a semiautomatic approach to evaluate a health indicator (HI) of an industrial freeze dryer is derived. The focus of [10] is on the freeze-dryer cleaning process—namely, cleaning in place (CIP)—and the used dataset is obtained from the water

Manuscript received 14 February 2023; revised 13 March 2023; accepted 13 March 2023. Date of publication 27 March 2023; date of current version 1 May 2023. This work was supported by GlaxoSmithKline (GSK) S.p.a. An earlier version of this paper was presented at the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Republic of Korea, April 2021 [DOI: 10.1109/ICAIIC51459.2021.9415206]. The associate editor coordinating the review of this article and approving it for publication was Prof. Pierluigi S. Rossi. *(Corresponding author: Eleonora Oliosi.)*

Eleonora Oliosi and Gianluigi Ferrari are with the Internet of Things Laboratory (IoT Lab), Department of Engineering and Architecture, University of Parma, 43121 Parma, Italy (e-mail: eleonora.oliosi@unipr.it; gianluigi.ferrari@unipr.it).

Gabriele Calzavara was with the Internet of Things Laboratory (IoT Lab), Department of Engineering and Architecture, University of Parma, 43121 Parma, Italy. He is now with Ammagamma S.r.l., 41121 Modena, Italy (e-mail: gabriele.calzavara1992@gmail.com).

Digital Object Identifier 10.1109/JSEN.2023.3260314

flow rate signal of a spray tube used in the CIP process. The density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm [11] is used to build a robust HI. In this article, we extend the analysis of the CIP process by considering other artificial intelligence (AI)-based data analysis strategies based on various clustering methods, namely, k-Means and Gaussian mixture models (GMMs). In addition to these two clustering methods, the principal component analysis (PCA) is applied for visualization purposes. We also investigate the applicability of other outlier detection algorithms, namely, the one-class support vector machine (SVM) and the local outlier factor (LOF). Moreover, a second process run in the freeze dryer—namely, the leak test (LT)—is considered for the derivation of an HI of other components of the freeze dryer. In this case, the relevant dataset for the LT is obtained from the pressure signal recorded during the LT process. In both CIP and LT cases, we show that DBSCAN outperforms other AI data analysis algorithms. All the considered algorithms are adopted for two operational approaches: a posteriori analysis and real-time monitoring of the evolution of the considered system health status. In the latter case, a possible approach to predictive maintenance is proposed.

This article is organized as follows. In Section II, the system background, in terms of the two freeze-drying analyzed processes, is presented. In Section III, a semiautomatic approach to inspect the water flow rate (in the CIP process) and the pressure (in the LT process) signals with the aim of computing an HI is illustrated, considering a posteriori analysis and real-time monitoring as possible operational approaches. In Sections IV and V, the obtained results for CIP and LT are presented, respectively. In Section VI, a discussion on the obtained results, failure modes, and possible extensions of our approach is presented. In Section VII, conclusions are drawn.

II. System Background

Freeze-drying, or lyophilization, is a process that involves three phases: 1) freezing the product; 2) lowering the pressure; and 3) removing the ice by sublimation based on temperature increase (primary drying and secondary drying). This process aims at drying the product by removing the water in it without damaging its qualities. In order for this to happen, the product is frozen to a temperature below its so-called eutectic point (i.e., the lowest freezing point of a mixture) [12], which must be carefully determined together with the freezing rate. As a matter of fact, a slow freezing rate will produce a more porous structure, characterized by a shorter sublimation rate but more difficult to reconstitute, whereas a fast freezing rate will result in a more granulated structure, easier to reconstitute but with a longer sublimation rate. Freeze-drying is largely used in the pharmaceutical field since its operational conditions guarantee that the final product, despite shape transformation, keeps all its initial qualities and preserves them over time. As a matter of fact, the preservation of the physicochemical properties is fundamental when dealing, for instance, with vaccines or genetic material.

Industrial freeze-drying takes place in machines denoted as *freeze dryers* or *lyophilizers*, which are designed to reach and maintain specific temperature and pressure conditions needed for the process to be successful. A lyophilizer is a steel machine consisting of two main chambers: the largest one contains the plates on which the product is positioned during the freeze-drying process; the smallest one contains a condenser, inside which liquid nitrogen flows at extremely low temperatures. In addition to the lyophilization cycle, other automated processes are run in the freeze dryer with the aim of cleaning, sterilizing, or testing its integrity. Among these additional processes, CIP and LT are of interest in this work.

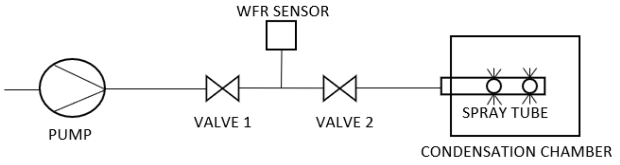


Fig. 1. Piping and instrumentation diagram (P&ID) of the water supply for the condenser spray tube.

A. CIP

CIP consists of cleaning the freeze dryer with purified water. In the reference machine in this work, five spray tubes are used to spray the chamber walls, the shelves, the condenser walls, and the condenser plates. Each spray tube has multiple nozzles that pour water into the machine. Four spray tubes are located in the main chamber, whereas there is only one in the condenser. The condenser spray tube is prone to strong mechanical and thermal stresses since it is used to spray hot water—as a matter of fact, the temperature gap between the steel of the condenser and the sprayed water can be as high as 150 °K. As a consequence, leaks in the spray tube can occur frequently, and the machine runs the risk of being washed incorrectly. As the freeze dryer must always be in sterile conditions, extreme attention has to be paid to cleaning. Therefore, one needs to regularly monitor the status of the spray tube in order to keep correct operational conditions. In Fig. 1, the components of the freeze-dryer watering system are shown only for the condenser, as it will be the subsystem of reference for CIP considered in the rest of this work.

During the CIP process, valve 1 stays open, while valve 2 opens and closes three times over a 50-s time interval. During this period, the water pushed by the pump flows through the nozzles and enters the condenser. This procedure is the same for the four spray tubes of the main chamber. A water flow rate sensor (WFRS), used to monitor the process, measures the water flow rate (dimension: [m^3/h]) through each spray tube. The WFRS signal, thus, allows for calculating the total amount of water that has been sprayed in the machine. The sampling rate of the WFRS is 1 sample/s.

A spray tube's conditions can only be monitored by the WFRS signal associated with the water flowing into the freeze dryer. During the CIP process, since the five spray tubes are turned on in disjoint time intervals, at each time instant, the WFRS signal is representative of the unique spray tube that is pouring water into the machine.

The water flow streamed from a spray tube is characterized by a rate that depends on two factors:

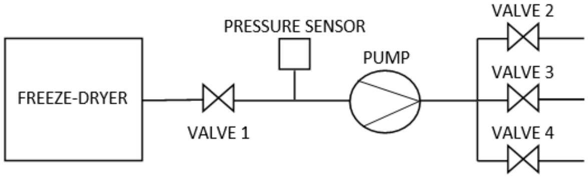


Fig. 2. P&ID of the components involved in the LT process, namely, a vacuum pump and four vacuum valves. The pressure sensor used to record the pressure signals is also highlighted.

1) the spray tube's structural (health) conditions, associated with the deterioration of its steel components;
2) the performance of the pump that pushes water to the nozzles, which can vary its thrust force depending on its health status.

For this work, an industrial freeze dryer located in the production plant of GlaxoSmithKline (GSK) in San Polo di Torrile (Parma, Italy) is considered. The used historical data are collected from all the CIP processes from November 2014 to November 2019.

B. Leak Test

LT is necessary to measure the sealing of the freeze dryer. In fact, because of strong thermal variations, microscopical cracks can appear, mostly in tubes and support structures. These cracks may create a leak, namely, an influx of gas into the drying chamber. As a consequence, the sterile product environment inside the chamber is contaminated, no matter the leak size, and the final products' quality is compromised (possibly leading to significant economic losses). The system components that are involved in the LT process are shown in Fig. 2. Essentially, during LT, all the border valves (valve 2, valve 3, and valve 4) of the freeze dryer are, first, closed. Then, the vacuum pump is activated, and valve 1 is opened. When the freeze dryer reaches the internal pressure of 10 μbar, valve 1 is closed, and the pressure increase is measured over a fixed time interval (generally 1.5 h). If the pressure increase is evaluated as anomalous, e.g., it becomes too high, LT is declared failed, and maintenance activity on the lyophilizer sealing is required.

The process signal that can be used to monitor the status of the freeze-dryer sealing is the pressure signal, which is shown in Fig. 3 (over 235 consecutive cycles from January 2016 to January 2020). This signal is extracted by the pressure sensor shown in Fig. 2. We underline that, in this case, the pressure increase measured during the LT process cannot be associated with a single component, but it depends on the health status of the multiple components that all contribute to lyophilizer sealing.

The analyzed historical data refer to the LT processes from January 2016 to January 2020 carried out at the freeze dryer mentioned for CIP at the end of Section II-A.

III. Proposed Data Analysis Approach

In this work, we propose an approach based on the analysis of data recorded by sensors installed on industrial machines and expedient to describe the operational conditions of the machine component of interest. Our goal is to estimate the health status of the considered component and, then, to predict its time evolution. To this purpose, the illustrated approach involves, first, the extraction of features and, then, the application of clustering in order to highlight, through proper time segmentation, the system status evolution over time. For this reason, we refer to our clustering-based approach as "time-aware." In Sections IV and V, it will be shown that DBSCAN, making use of the analyzed process cycle number as a fundamental feature, is the best clustering method to be adopted for data analysis. As a final step, an HI will be derived from the clustered data. As anticipated, two operational approaches will be considered: a posteriori analysis and real-time monitoring. In the second case, a predictive model is proposed in order to identify anomalous variations of the considered system health status, thus enabling predictive maintenance. In the remainder of this section, we sketch the main "ingredients" of our approach.

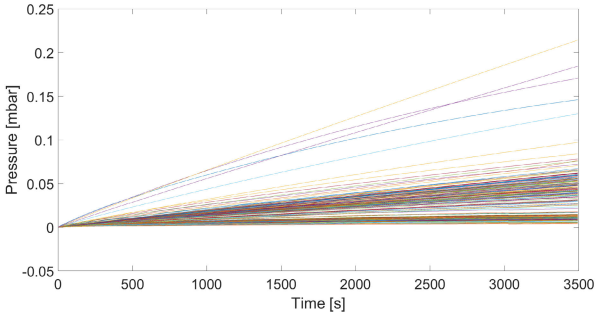


Fig. 3. LT pressure signal extracted by the pressure sensor shown in Fig. 2 over 235 consecutive cycles from January 2016 to January 2020.

The extraction of statistical features from time-domain sensor signals and the computation of the monotonicity as a features' selection method have been proposed in the literature for health status monitoring [13], [14]. The main novel contributions of this article are given as follows:

1) the median-based aggregation method in the signal processing approach and the HI evaluation to obtain more robust results;
2) the use of a "time-aware" clustering to study the time evolution of the health status of the considered component;
3) the computation method for the HI after the DBSCAN-based outlier removal;
4) the predictive approach based on a linear interpolation of the real-time computed HI.

A. Single Sensor Signal Processing

The HI is a time-dependent indicator that describes the evolution (namely, the degradation) of the industrial machine under analysis, more precisely of one of its components [15]. A key role in HI computation is played by the features chosen to describe the signals recorded by the component's sensors. This choice can heavily affect HI evaluation and accuracy. However, features' selection cannot abide by any a priori rules since the specific scenario of interest and the knowledge of the considered process must be taken into account. As a matter of fact, given that the signals are possibly heterogeneous, the associated descriptive information needs to be extrapolated accordingly. Being the sensor signals usually affected by noise, "smoothing" can be applied to better highlight the underlying trend of the extracted features, as suggested in [16]. In order to do this, a causal moving median filter with a window of six taps[1] is used, resulting in the following smoothed signal (associated with the most recent time epoch of the window):

$$f_{\text{smooth}}(i) = \begin{cases} \text{median}[f(i-5), \dots, f(i-1), f(i)] \\ \qquad\qquad\qquad\qquad 6 \leq i \leq N \\ \text{median}[f(1), \dots, f(i)], \quad 1 \leq i < 6 \end{cases} \tag{1}$$

where $f(i)$ is the value of feature f in the ith cycle ($i = 1, \dots, N$) and N represents the number of all available cycles.

B. Monotonicity of Sensor Signal Features

Once the features are extracted and smoothed, their "potential" to predict the deterioration of the machine component must be evaluated. This potential can be quantified in terms of monotonicity, defined as follows:

$$\text{monotonicity}(f, N) \triangleq \left| \sum_{i=1}^{N-1} \frac{\text{sgn}\left[f_{\text{smooth}}(i+1) - f_{\text{smooth}}(i)\right]}{N-1} \right| \tag{2}$$

where $f_{\text{smooth}}(i)$ is the value of smoothed feature f in the ith cycle [defined in (1)] and $\text{sgn}[n] = \pm 1$ if $n \gtrless 0$, respectively [16]. The monotonicity value always belongs to the interval [0, 1] and assesses how well a feature is representative of the system evolution: the closer the monotonicity value to 1, the more representative the feature.

In [16], other features' selection methods used for predictive maintenance are proposed, such as prognosability and trendability. *Prognosability* measures the variance in the critical failure value of a population of systems: namely, it measures the variability of the indicators at failure. *Trendability* indicates the similarity between the trajectories, measured in several run-to-failure experiments, of a feature. Trendability is useful to determine which indicator best tracks the degradation process since the most "trendable" feature tends to always have the same behavior when the analyzed system gets progressively closer to failure. Even though prognosability and trendability are meaningful criteria, the chosen features' selection criterion is monotonicity because, by means of a monotonous feature, a failure in the described process can be instantly identified. As a matter of fact, if there is an abrupt change in the underlying trend of the feature, it is intuitive to conclude that something anomalous has occurred to the considered machine component since degradation is typically an irreversible process. Moreover, in order to adopt the prognosability and trendability selection methods, a much larger quantity of data describing a failure of the considered machine component would be required.

In this work, the monotonicity is computed on the features extracted from a training dataset, which includes 40% of the whole available dataset. In fact, in this way, the monotonicity results can be used also for real-time monitoring. In order to obtain more robust monotonicity measurements, before applying (sliding window-based) smoothing, the extracted features' values are aggregated in consecutive and disjoint groups of n_{median} elements, and the median of each group is computed. For instance, other results (not shown here for lack of space) obtained by computing the arithmetic average instead of the median show that the median is the most effective aggregation method. The overall signal processing strategy is shown in Fig. 4, with n_{median} set illustratively to 5.

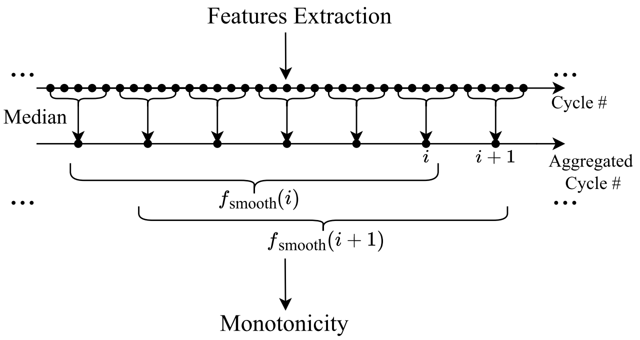


Fig. 4. Signal processing steps: from feature extraction to the monotonicity computation ($n_{\text{median}} = 5$).

C. DBSCAN-Based Clustering

As anticipated, the classification of the machine operational conditions revolves around clustering, based on the use of DBSCAN, of a properly extracted feature of the sensed signal. The clustered data lead naturally to a time series segmentation. The proposed approach could make use, in the place of DBSCAN, of other clustering and outlier removal algorithms applicable to our problem. In Sections IV-C and IV-D, V-C, and, more generally, VI-A, relevant comparisons among DBSCAN and other algorithms are carried out.

DBSCAN is a clustering algorithm relying on a density-based notion of clusters. In [11], it is stated that "the key idea is that, for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points, i.e., the density in the neighborhood has to exceed some threshold." As a matter of fact, two parameters are required: 1) the minimum number of items per cluster, denoted as "minPts" and 2) the distance ϵ corresponding to the radius of a neighborhood of a given point in the cluster. The value of ϵ is estimated through different steps: 1) for each point in the input database, the distance to the minPtsth nearest point is evaluated; 2) a graph is generated after sorting, in ascending order, the points according to the computed distance values; and 3) an "elbow" in this graph can be identified, and its corresponding distance is chosen as ϵ. The criterion for the choice of the minPts value is that it must be a number larger than or equal to one plus the number of dimensions of the input data in the features' space. This criterion has been derived from [11], where it is shown that, for a 2-D input database, the minPts-distance graph with minPts = 4 is not significantly

[1] Our results show that using six taps provides a good compromise between complexity and performance.

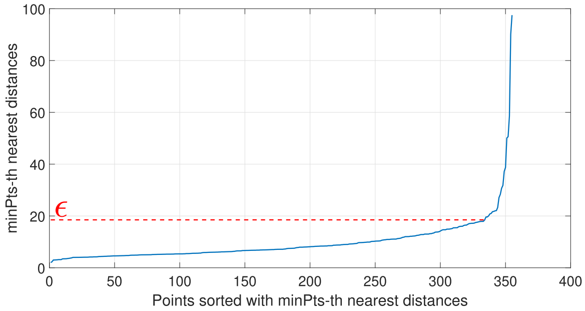


Fig. 5. minPts-distance graph (minPts = 5) used for CIP process (see Section IV-B for more details). The corresponding value of ϵ is indicated.

different from the ones obtained with larger values of minPts, while being computationally more efficient.

In Fig. 5, we show the minPts-distance graph obtained in the case of the CIP process (see Section IV-B1 for more details). On the x-axis, the points of the input database, sorted in ascending order of their computed minPtsth nearest distance, are indicated. With minPts = 5, it can be observed that $\epsilon \simeq 17$. As a matter of fact, for $\epsilon > 17$, the points start becoming noisy.

Given ϵ and minPts, DBSCAN allows to identify three kinds of point [17].

1) *Core Point:* A point in a cluster that has at least minPts points in its ϵ-neighborhood.
2) *Border Point:* A point in a cluster that has a number of points in its ϵ-neighborhood smaller than minPts but larger than one.
3) *Noise Point:* A point that in its ϵ-neighborhood has only one point, i.e., itself.

Three concepts turn out to be fundamental for DBSCAN: 1) direct density reachability; 2) density reachability; and 3) density connectivity. A point p is *directly density-reachable* from a point q, with respect to ϵ and minPts, if p belongs to the ϵ-neighborhood of q, and the cardinality of the ϵ-neighborhood of q is larger than or equal to minPts. A point p is *density-reachable* from a point q, with respect to ϵ and minPts, if there is a chain of points $p_1, \ldots, p_m$, with $p_1 = q$ and $p_m = p$, such that p_{i+1} is directly density-reachable from p_i, $i = 1, \ldots, m-1$. A point q is *density-connected* to a point p, with respect to ϵ and minPts, if there is a point r such that both q and p are density-reachable from r with respect to ϵ and minPts [11].

At this point, the steps involved in DBSCAN-based clustering can be summarized as follows [17].

1) A point p is randomly chosen, and all points density-reachable from p, with respect to ϵ and minPts, are retrieved. At this point, there are two possibilities.
   a) If p is a core point, a cluster with respect to ϵ and minPts is formed.
   b) If p is not a core point and no points are density-reachable from p, then the algorithm passes to the next data point by identifying p as a noise point.
2) If a cluster is fully expanded (all points within reach are visited), then DBSCAN proceeds to iterate through the remaining unvisited points in the dataset.

As will be shown in Sections IV-A and V-A, only one of the features extracted from the recorded signals for each process (either CIP or LT) will be sufficiently monotonous to be considered for the HI derivation. This feature's values will then be used as input to DBSCAN together with the number of the analyzed process cycle in order to obtain a "time-aware" data clustering. For the purpose of making these two features comparable, as an initial step, we z-score normalize (by subtracting the mean, over all available observations, from the value at each epoch and dividing this difference by the standard deviation [18]) the resulting most monotonous feature. Subsequently, we multiply it by a constant (heuristically selected) equal to 100 to obtain approximately the same order of magnitude as the cycle numbers.

The main benefits of the DBSCAN-based clustering algorithm, with respect to other methods illustrated in the literature and discussed later, are given in the following.

1) The amount of clusters is not to be set before the algorithm application.
2) The outliers are automatically identified.

D. Health Indicator

In general, there is no fixed rule for the computation of the HI. We now propose a novel method, developed through successive refinements, while analyzing the available data and the obtained results.

As mentioned in Section III-C and as will be shown in Section IV-A, only one feature, after monotonicity evaluation, will be selected for the HI computation, during the CIP process, given the high correlation among the resulting three most monotonous features (the same will happen in Section V-A for the LT process). After removing the outliers found by means of DBSCAN, as described in Section III-C, and according to the aggregation approach proposed in Section III-B, these feature values are divided in consecutive and disjoint groups[2] of $n_{\text{median}} = 3$ elements (associated with three consecutive cycles). Then, the median of each group is calculated (in order to obtain the aggregated most monotonous feature value representative of that group). At this point, smoothing is applied according to (1). Finally, the HI is evaluated by relying on the aggregated and smoothed feature values. According to this approach, the HI will be represented as a function of the aggregated cycle number, which is derived from the considered process cycles according to the aggregation factor n_{median} and the considered available cycle set. In Section III-E, two operational approaches will be considered, namely, *a posteriori analysis* and *real-time monitoring*: in both cases, $n_{\text{median}} = 3$. However, for *a posteriori analysis*, the available cycle set to which n_{median} is applied will be equal to each manually identified cluster, whereas, for *real-time monitoring*, it will be equal to the interval of ten process cycles. More details will be provided later.

Should at least two uncorrelated (or weakly correlated) features be the most monotonous ones, a multidimensional

[2] In this case, we select consecutive and disjoint groups of three elements, instead of five, as for the monotonicity calculation mentioned in Section III-B and represented in Fig. 4, since the cardinality of the available process cycles set is much smaller than the cardinality of an entire dataset.

extension of our approach would be required. This will be discussed in Section VI-D.

E. Operational Approaches

The four operational steps discussed above—namely, the single sensor signal processing (see Section III-A), the computation of the monotonicity of the extracted features (see Section III-B), the DBSCAN-based clustering (see Section III-C), and the HI evaluation (see Section III-D)—can be used for both a posteriori analysis of the system health status evolution over time (as off-line data analysis) and real-time health status monitoring (for the purpose of predictive maintenance).

1) A Posteriori Analysis: In order to obtain a posteriori overview of the machine health status and HI evolution, one can consider all the available sensor data. Although the results are not useful for practical (maintenance) purposes, they allow one to obtain a posteriori evaluation in terms of both data clustering and HI. For DBSCAN-based clustering, we use as input the considered process (either CIP or LT) cycle number and the resulting most monotonous feature's values collected over all the available cycles of the considered process, properly processed as described in Section III-C. As for the HI computation, we introduce a preparatory step to obtain more robust results. This step involves the identification of the pauses between two consecutive processes (either CIP or LT) cycles. We consider the time difference between two consecutive cycles, and we choose a threshold (namely, 300 h for CIP and 1500 h for LT, as will be discussed in Sections IV-B1 and V-B1, respectively), to select the most significant interruptions. These interruptions identify the separations between clusters to be identified. We call this process "manual clustering" to distinguish it from the automatic clustering provided by DBSCAN. At this point, the procedure for the HI derivation described in Section III-D is applied with aggregation and smoothing of the resulting most monotonous feature performed within each manually identified cluster.

The use of all the available (historical) sensor data is important also to analyze the extracted single sensor features. As a matter of fact, as mentioned in Section III-B, the monotonicity is computed on a dataset, including 40% of the whole available data. From a practical point of view, this can be considered a training step that returns the features to be used for real-time monitoring, making the presented real-time monitoring results meaningful.

2) Real-Time Monitoring and Application to Predictive Maintenance: For real-time monitoring, we choose to check the considered machine component health status every ten process cycles (either CIP or LT). In particular, up to every ten process cycles: 1) the DBSCAN-based clustering is applied, as mentioned in Section III-C; 2) the outliers are identified and removed; and 3) the HI is computed, as described in Section III-D [with the most monotonous feature's values being aggregated by $n_{\text{median}} = 3$ elements and then smoothed according to (1)].

At this point, various methods, based on the analysis of the real-time monitoring results, can be used for the purpose of predictive maintenance. In this article, we present an approach based on endpoint linear interpolation of the real-time HI to identify potentially anomalous system behaviors. The endpoints are the HI value in correspondence to the starting cycle (or reset cycle) and the HI value of the last considered aggregated cycle, namely, $(t_{\text{in}}, \text{HI}_{\text{in}})$ and $(t_{\text{fin}}, \text{HI}_{\text{fin}})$. The straight line $\widehat{\text{HI}}(t) = at + b$ passing though the considered endpoints can be derived from the following expression:

$$\frac{t - t_{\text{in}}}{t_{\text{fin}} - t_{\text{in}}} = \frac{\text{HI} - \text{HI}_{\text{in}}}{\text{HI}_{\text{fin}} - \text{HI}_{\text{in}}} \tag{3}$$

from which

$$\begin{aligned} a &= \frac{\text{HI}_{\text{fin}} - \text{HI}_{\text{in}}}{t_{\text{fin}} - t_{\text{in}}} \\ b &= \text{HI}_{\text{in}} - \frac{t_{\text{in}}(\text{HI}_{\text{fin}} - \text{HI}_{\text{in}})}{t_{\text{fin}} - t_{\text{in}}}. \end{aligned} \tag{4}$$

The interpolating line $\widehat{\text{HI}}(t)$ intuitively needs to be compared with the effective value $\text{HI}(t)$ for $t = t_{\text{fin}} + 1$ (in general, $t > t_{\text{fin}}$): if the value of $\text{HI}(t)$ is sufficiently close to $\widehat{\text{HI}}(t)$, then one can conclude that there is no anomaly. In order to automatize the detection of anomalies, we consider the following two alarm threshold lines $\text{HI}^{(\pm\Delta)}$:

$$\begin{aligned} \text{HI}^{(+\Delta)}(t) &\triangleq at + b + \Delta \\ \text{HI}^{(-\Delta)}(t) &\triangleq at + b - \Delta \end{aligned} \tag{5}$$

where the value of Δ is specific for each monitored process. By trial and error, our results show that effective values are 0.25 m^3/h for CIP and 0.005 mbar for LT. If the HI computed in the next ten cycles is included between the lines $\text{HI}^{(\pm\Delta)}$, then the analyzed component operational conditions are considered correct. The interpolation slope a and intercept b values can then be updated taking into account the new cycles. On the contrary, if the next computed HI is outside the range between the lines $\text{HI}^{(\pm\Delta)}$, then an alarm can be emitted. This can be summarized as follows:

$$\begin{cases} \text{HI}^{(-\Delta)}(t) < \text{HI}(t) < \text{HI}^{(+\Delta)}(t), & \text{correct} \\ \text{HI}(t) < \text{HI}^{(-\Delta)}(t) \text{ or } \text{HI}(t) > \text{HI}^{(+\Delta)}(t), & \text{anomalous.} \end{cases} \tag{6}$$

Once an anomaly has been detected and the corresponding problem (if any) solved (through a predictive maintenance approach), the linear interpolation-based procedure can start again from the new cycle (after the problem solution) in order to detect the next future anomaly.

Together with HI interpolation and prediction, real-time data clustering can also highlight that something occurred to the considered system. As a matter of fact, if at a check point (namely, every ten process cycles) only one cluster is identified, this means that no variation appeared in the operational conditions of the considered component—the HI can start drifting but still remain within the alarm threshold range. On the opposite, when, at the check point, at least two clusters appear, this likely means that an anomalous event has taken place in the system (possibly a failure). In correspondence to the appearance of at least two clusters, the HI shows an abrupt deviation and exits out of the alarm threshold range.

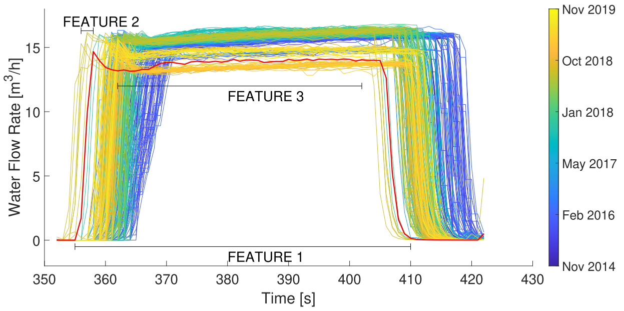


Fig. 6. WFRS signal and the three extracted intuitive (temporal) features.

In this work, we perform real-time monitoring with the same sensor data of the a posteriori analysis. In practical use cases, real-time monitoring would be based on new data (after a training phase carried out on the available data).

IV. Cleaning in Place

In this section, the proposed data analysis approach is applied to the sensor signals recorded during the CIP process. In Section IV-A, the single sensor signal processing and the features' monotonicity computation are described. In Section IV-B, DBSCAN-based a posteriori analysis and real-time monitoring are illustrated. In Section IV-C, other two clustering algorithms (namely, k-Means and GMM) and PCA are applied to our problem, as alternatives to DBSCAN, and their performances are investigated. In Section IV-D, other outlier removal algorithms (namely, one-class SVM and LOF) are considered as alternatives to DBSCAN, and a comparison among the obtained results is provided.

A. Single Sensor Signal Processing and Features' Monotonicity

In order to analyze the CIP process and, in particular, the time evolution of the health status of the components of the condenser water distribution system, the considered sensor is the WFRS, as mentioned in Section II-A. We follow the procedures for WFRS signal processing and features' monotonicity computation presented in Sections III-A and III-B, respectively.

With the purpose of computing the HI and detecting the considered system anomalies, three intuitive (temporal) features are extracted from the WFRS signal. Such features are illustrated in Fig. 6 and can be described as follows.

1) Feature 1 is the time interval between the instant at which the water flow rate goes above the threshold of 0.3 m^3/h and the instant at which it returns below this threshold.
2) Feature 2 quantifies the time taken by the spray tube to reach the maximum water flow rate: it coincides with the time interval between the instant at which the water flow rate becomes higher than 0.3 m^3/h and the instant at which it reaches the "steady-state" value (above 12 m^3/h).
3) Feature 3 corresponds to the WFRS signal average value during "steady-state" conditions (namely, the time interval during which the water flow rate is approximately maximum). For the purpose of computing this feature, the identification of a midpoint (MP) between the instant at which the water flow rate overcomes 12 m^3/h and the instant corresponding to its return below this threshold is required. Subsequently, the average value of the WFRS signal in an interval equal to 40 s centered at the MP is evaluated.[3]

TABLE I
Monotonicity of the Three Features of the WFRS Signal

	Feature 1	**Feature 2**	**Feature 3**
Monotonicity	0.01	0.01	0.56

The monotonicity of the three (temporal) features described above can be evaluated following the procedure outlined in Fig. 4 and according to (1) and (2). In particular, the total number of cycles is 355, and the number of aggregated cycles is $N = 355/n_{\text{median}} = 355/5 = 71$. Table I reports the obtained results: Feature 3 turns out to be the only feature with a sufficiently high monotonicity value (equal to 0.56).

In order to extend the approach outlined above, we extract other (common) statistical features from the WFRS signal: they are listed in Table II. In this table, $\{v(t)\}_{t=1}^{n}$ coincides with the WFRS signal in the (n-sample) time interval (per cycle) during which Feature 3 is evaluated. More precisely, the n samples are extracted from the 40-s time interval introduced above in the description of Feature 3. Since the WFRS sampling rate is 1 sample/s, it follows that $n = 41$.[4]

The statistical features listed in Table II are computed for each CIP cycle. Their monotonicity is computed by following the steps outlined in Fig. 4. The monotonicity values of the computed features (namely, all statistical features in Table II; Feature 1 and Feature 2) are shown in Fig. 7. One can observe that almost all monotonicities are below 0.2 but for "Mean" (corresponding to Feature 3 in Table I), "rms," and "squared factor." Since these three features turn out to be correlated by 99%, one of them can be selected as representative of the remaining two. As a consequence, we select "mean" (i.e., Feature 3) as the only relevant feature of the WFRS signal. In the rest of this article, it will be referred to as "FlowMean."

B. DBSCAN-Based Analysis and Monitoring

As mentioned in Section III-C, DBSCAN is applied to two features:

1) FlowMean;
2) the CIP process cycle number (denoted as CycleNumber).

The "heuristic" transformation described in Section III-C makes the value of FlowMean numerically comparable to that of CycleNumber.

1) A Posteriori Analysis: For the a posteriori analysis, we follow the approach discussed in Section III-E1.

[3] The time interval during which the water flow rate is maximum is larger than 40 s.

[4] The total number of samples is 41 because the first considered sample is at 0 s and the last considered sample is at 40 s.

TABLE II

TIME-DOMAIN STATISTICAL FEATURES OF A TIME-DISCRETE SIGNAL $\{v(t)\}_{t=1}^{n}$. IN THIS WORK, $\{v(t)\}_{t=1}^{n}$ COINCIDES WITH THE WFRS SIGNAL IN THE (n-SAMPLE) TIME INTERVAL DURING WHICH THE WATER FLOW RATE IS MAXIMUM. THE MEAN CORRESPONDS TO FEATURE 3

Feature	Mathematical expression
Mean	$\mu = \frac{1}{n}\sum_{t=1}^{n} v(t)$
Standard Deviation (Std)	$\sigma = \sqrt{\frac{1}{n}\sum_{t=1}^{n}[v(t)-\mu]^2}$
Skewness	$\frac{\frac{1}{n}\sum_{t=1}^{n}[v(t)-\mu]^3}{\sigma^3}$
Kurtosis	$\frac{\frac{1}{n}\sum_{t=1}^{n}[v(t)-\mu]^4}{\sigma^4}$
Peak2Peak	$\max_{t=1\ldots n}\{v(t)\} - \min_{t=1\ldots n}\{v(t)\}$
Root Mean Square (RMS)	$\sqrt{\frac{1}{n}\sum_{t=1}^{n} v^2(t)}$
Crest Factor	$\frac{\max_{t=1\ldots n}\{v(t)\}}{RMS}$
Shape Factor	$\frac{RMS}{\sum_{t=1}^{n}\lvert v(t)\rvert/n}$
Impulse Factor	$\frac{\max_{t=1\ldots n}\{v(t)\}}{\sum_{t=1}^{n}\lvert v(t)\rvert/n}$
Margin Factor	$\frac{\max_{t=1\ldots n}\{v(t)\}}{\left[\sum_{t=1}^{n}\lvert v(t)\rvert/n\right]^2}$
Squared Factor	$\sum_{t=1}^{n} v^2(t)$

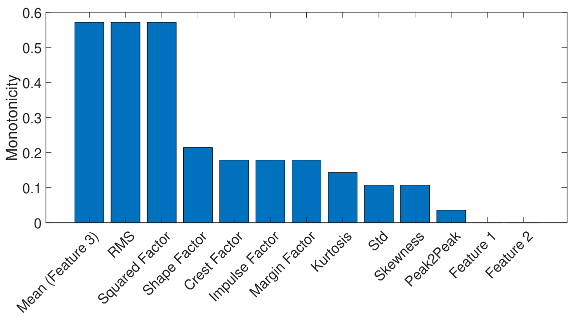


Fig. 7. Monotonicity of the extracted features (all statistical features of WFRS: Feature 1 and Feature 2).

The chosen (minPts, ϵ) configuration for DBSCAN is (5,17), as can be derived from the minPts-distance graph in Fig. 5. We set minPts to 5 because it is a reasonable value according to the general rule discussed in Section III-C, and it turns out to be the most suitable value for our application in terms of data clusters' identification (minPts $= 4$ worsens the performance). The outcome of the DBSCAN-based data clustering is shown in Fig. 8. It is important to remark that the borders between adjacent clusters correspond to modifications carried out in the system (e.g., maintenance acts). However, it can be noticed that cluster 3 behaves in an unexpected way since it is identified in the same time interval as cluster 2, despite being represented by different FlowMean values. Although the motivation behind this anomalous behavior is likely associated with physical conditions, it is noteworthy that it can be detected by an automatic data clustering method—this will be investigated in Section IV-B2. As one can observe in Fig. 8, DBSCAN can also discriminate, together with the anomalous cluster, all the isolated points, identifying them as outliers. We remark that no ground truth is available for outlier detection since the only available auxiliary information is about the four maintenance acts on the considered system. Based on this information, the data clustering in Fig. 8 is as accurate as possible, and consequently, we rely on the same data for outlier identification.

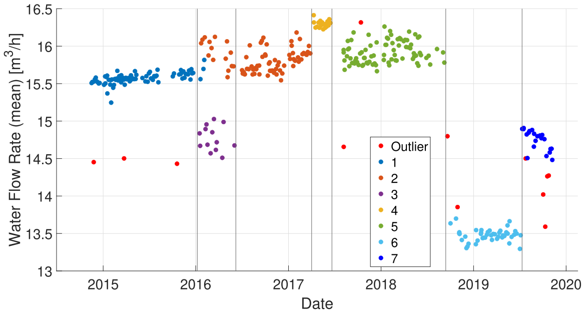


Fig. 8. CIP process: DBSCAN-based computed clusters (minPts $= 5$ and $\epsilon = 17$) [10].

At this point, we derive an HI following the procedure described in Section III-D after a "manual clustering" preparatory step. The "manual clustering" results are shown in Fig. 9, where the outliers and the anomalous clusters found by DBSCAN have been removed from the data. The time threshold is set to 300 h. It can be noticed that: 1) not all the interruptions coincide with maintenance acts that modify the water flow rate values and 2) the chosen time threshold does not allow to identify the event that determines the transition from cluster 6 to cluster 7 in Fig. 8 (this maintenance lasts much less than 300 h). The resulting HI of the components contributing to the water distribution in the condenser is shown in Fig. 10.

The maintenance acts carried out on the pump are highlighted as vertical green lines. More precisely, the first two maintenance acts are the occurrences determining the cluster 4 boundaries in Fig. 8—the exact same cluster can also be observed in Fig. 9. The first maintenance act (March 27, 2017) alters the status of the system, and then, the second one (June 15, 2017) makes it return to its initial condition. Only one maintenance act was performed on the spray tube (identification and subsequent weld of a leak) in the considered time interval: this is represented by the red vertical line at

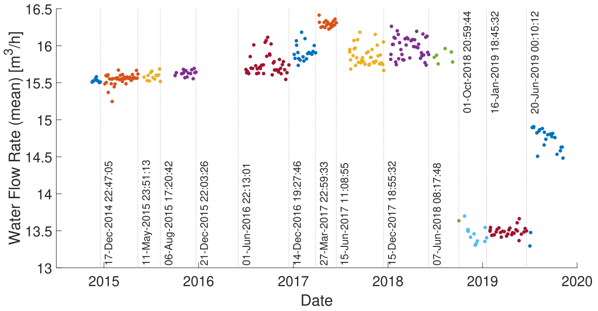


Fig. 9. CIP process: “manual” clustering (time threshold = 300 h).

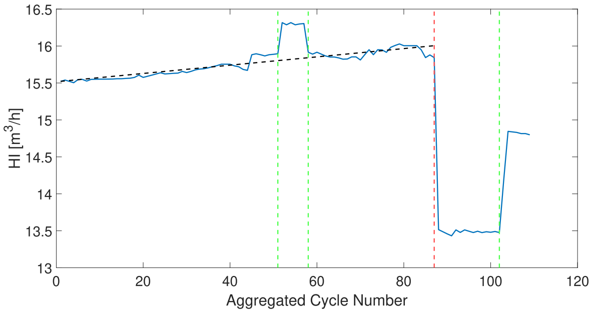


Fig. 10. HI of the condenser water distribution system.

cycle 88 in Fig. 10. It can be observed that the HI grows linearly from the initial data instant to the welding date, except for the interval between the first two pump maintenance acts, in correspondence to which it deviates. This observation is the basis for our predictive model, discussed in Section III-E2 and applied to this problem in Section IV-B2. The green line at cycle 104 refers to a process modification of the water distribution system (third pump maintenance act), which intentionally leads to an increase in water flow rate from that cycle onward.

2) Real-Time Monitoring and Application to Predictive Maintenance: In order to monitor, in real time, the health status of the condenser water distribution system, the approach discussed in Section III-E2 is followed. For illustrative purposes, we will show only the results up to the following CIP cycle numbers: 190, 300, and 340. This choice is motivated by the fact that these cycles belong to the cycle intervals (the results are updated every ten process cycles) immediately after maintenance acts on the considered system. As for the DBSCAN-based clustering, the same (minPts, ϵ) configuration used for the a posteriori analysis (i.e., minPts $= 5$ and $\epsilon = 17$) is adopted, and the obtained results are shown in Figs. 11(a), 12(a), and 13(a), respectively. It can be observed that, in all these cases, the cluster identified after the maintenance act is different from the cluster identified before it. Therefore, DBSCAN is able to track the variations of the component health status in real time. Moreover, in Fig. 11(a), it can be noticed that all the anomalous cycles at the beginning of the year 2016 are identified as outliers (as further confirmation, see Fig. 14). Only at a later stage, when multiple years of data are processed, these cycles are recomputed as two clusters (namely, part of cluster 2 and cluster 3) in Figs. 12(a) and 13(a). Nevertheless, from an operational point of view, this phenomenon does not affect the performance of the proposed approach since the focus of real-time monitoring is on the current health status variation.

The HI evolution, associated with clustering, up to cycles 190, 300, and 340 is shown in Figs. 11(b), 12(b), and 13(b), respectively. The alarm thresholds are set considering $\Delta = 0.25$ m^3/h in (5). It can be observed that the HI computed in real time represents all the condenser water distribution system health status variations over time, which occur up to the check points, namely, the pump maintenance acts (March 2017 and June 2019) in Figs. 11(b) and 13(b), and the weld of the spray tube leak (October 2018) in Fig. 12(b). It can be noticed that our predictive approach, based on interpolation of the endpoints of the real-time HI, is effective for anomaly detection. As a matter of fact, under all the reported real-time monitoring circumstances, the HI overcomes the predicted alarm thresholds only in correspondence to the repairs activities that, in fact, modify the system’s health status.

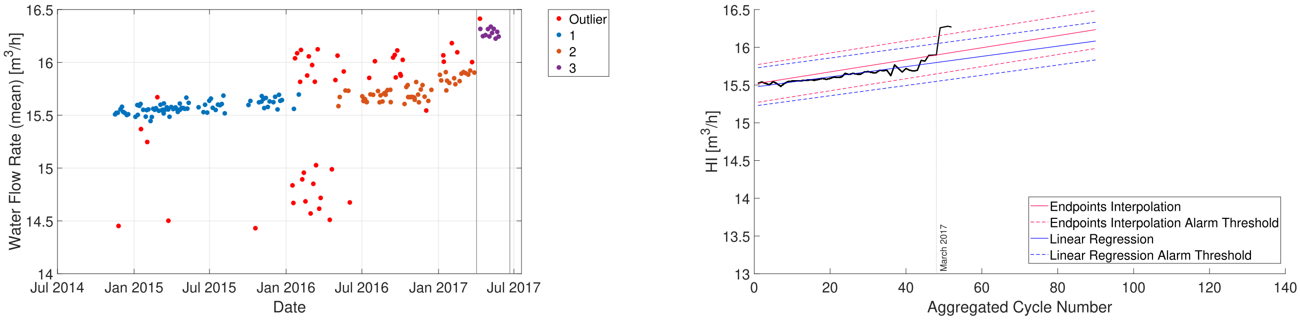
In Fig. 11(b), we compare our method results with the ones obtained with a least-squares linear regression. It can be observed that, even if interpolation- and linear regression-based results are slightly different, the outcome, in terms of prediction of the real-time system status variation, is the same since, in both cases, the HI overcomes the alarm thresholds when the activity on the pump is carried out. Therefore, in the remainder of this work, we will consider the interpolation-based approach proposed in (3)–(5), being computationally simpler.

As noticed with the DBSCAN-based clustering, the anomalous cycles at the beginning of the year 2016 are considered as outliers when monitoring, in real time, this time period [see Figs. 11(a) and 14] but are then recomputed as clusters when many more data are processed [see Figs. 12(a) and 13(a)]. This phenomenon can also be observed in the HI evaluation. As a matter of fact, by comparing Fig. 11(b), obtained up to cycle 190 in 2017, and Fig. 13(b), obtained up to cycle 340 in 2019, many more oscillations can be observed in the latter real-time HI in the interval between the aggregated cycles 20 and 40: this is due to the fact that the anomalous cycles are now taken into account in the calculations. This phenomenon does not affect, from an operational point of view, the performance of the proposed method also in this case since the focus is now on the predicted HI behavior and not on the past events.

C. DBSCAN Versus k-Means, GMMs, and PCA

In order to motivate the selection of DBSCAN, we compare its performance with those of other two relevant clustering algorithms, namely, k-Means and GMMs. PCA is also performed in order to better visualize the data clustered structure.

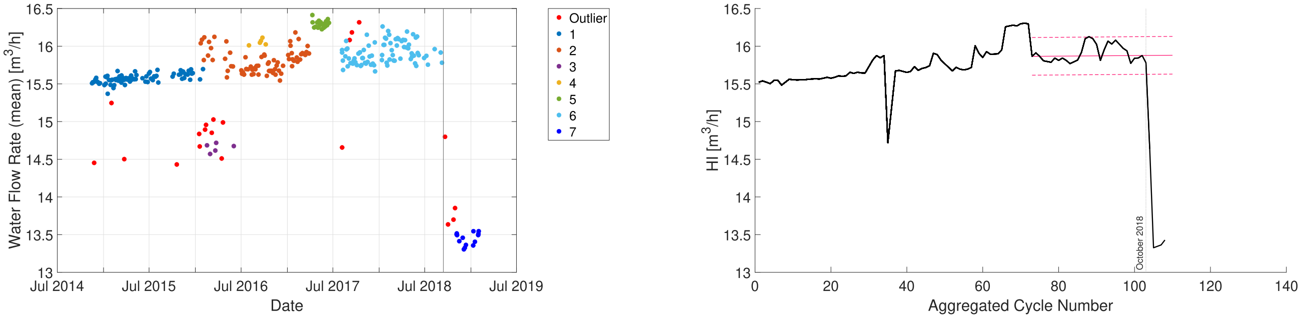
k-Means partitions a set of n_{obv} observations into k clusters so that the intercluster similarity is minimized and the intracluster similarity is maximized. The similarity is expressed in terms of the mean value of the observations in a cluster [19]. As a matter of fact, each data item is assigned to its most similar cluster, namely, the cluster where the distance between the item itself and the mean value of all the currently present cluster items (denoted as cluster centroid) is minimum. Unlike DBSCAN, the number of clusters k must be set a priori. This is



(a)

(b)

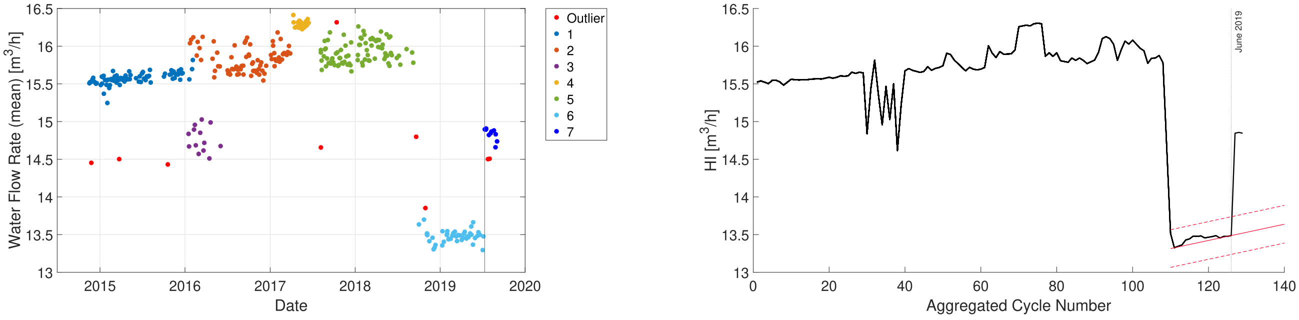
Fig. 11. Real-time monitoring up to CIP cycle 190: (a) DBSCAN-based clustering (minPts = 5 and ϵ = 17) and (b) HI.



(a)

(b)

Fig. 12. Real-time monitoring up to CIP cycle 300: (a) DBSCAN-based clustering (minPts = 5 and ϵ = 17) and (b) HI.



(a)

(b)

Fig. 13. Real-time monitoring up to CIP cycle 340: (a) DBSCAN-based clustering (minPts = 5 and ϵ = 17) and (b) HI.

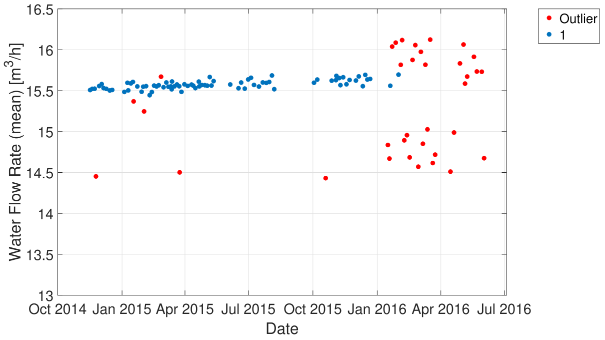


Fig. 14. DBSCAN-based clustering (minPts = 5 and ϵ = 17) until mid-2016.

critical for the application at hand, especially when the amount of data to be clustered keeps on increasing with real-time data acquisition, and the number of clusters is expected to increase over time.

GMMs are a family of distribution-based clustering algorithms. A GMM assumes that the data points have a Gaussian distribution. The shape of the clusters, the so-called "components," is determined by two parameters, namely, the mean and the standard deviation of the distribution [20]. As k-Means and unlike DBSCAN, the number of components must be set a priori by the user. This represents a disadvantage for the application of a GMM to our scenario because the number of machine statuses is not known in advance.

Therefore, since both k-Means and GMM require the number of clusters to be set a priori, for the purpose of a fair comparison, in Section IV-C1, as for the a posteriori analysis considered in Section IV-B1, the same number of clusters

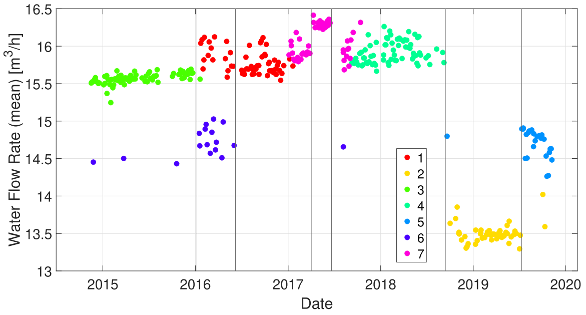


Fig. 15. CIP process: k-Means-based computed clusters ($k = 7$).

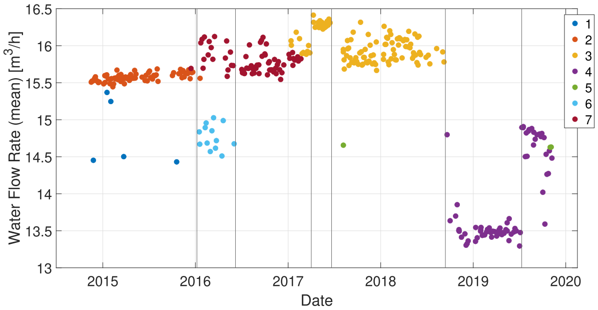


Fig. 16. CIP process: GMM-based computed clusters (set a priori to 7).

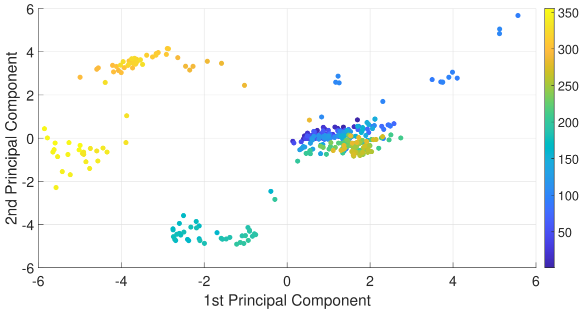


Fig. 17. CIP process: PCA-based computed clusters.

identified by DBSCAN in Section IV-B1 is used in k-Means and GMM. On the other hand, in Section IV-C2, as for real-time monitoring in Section IV-B2, an approach based on the identification of two consecutive clusters is proposed in order to verify the validity of these clustering algorithms for predictive maintenance (i.e., to detect a single change of status).

1) A Posteriori Analysis: In Fig. 15, the clusters identified by k-Means are shown. It can be observed that, for $k = 7$, i.e., for a value of k equal to the number of clusters found by DBSCAN in Fig. 8, the detected clusters differ from the ones predicted by DBSCAN. In particular, it can be noticed that cluster 4 in Fig. 8 is not correctly detected by k-Means; rather, it is included in cluster 7 of Fig. 8 with many other cycles belonging to the two adjacent statuses. This highlights a major problem of k-Means: if the clusters representing the machine statuses have very different sizes (in terms of the number of cycles), k-Means cannot separate data correctly. Moreover, unlike DBSCAN, k-Means cannot automatically identify the outliers. Cluster- or distance-based methods, which allow removing the outliers and can be used together with k-Means, have been proposed [21]. Nevertheless, using these methods requires setting additional parameters, such as the cardinality of the k-nearest neighbors set.

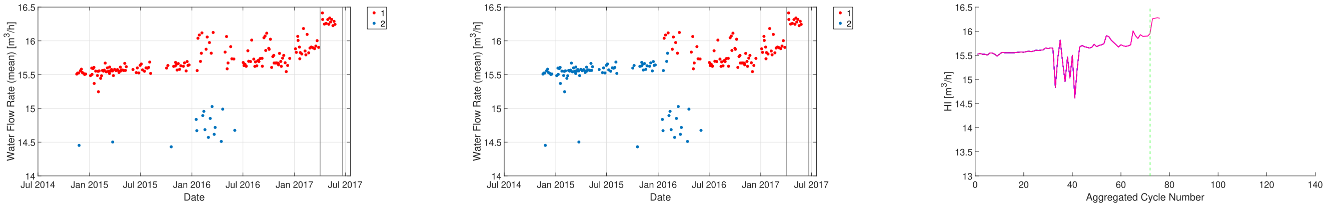
As for GMM, for a fair comparison with DBSCAN and k-Means, the number of components to be found is set to 7, as anticipated above. From the results shown in Fig. 16, it can be observed that GMM has worse performance, in terms of status identification, than DBSCAN. For instance, the cluster included between the first two pump maintenance acts—cluster 4 in Fig. 8—is not correctly identified. Moreover, clusters 6 and 7 in Fig. 8 are now merged together into a single cluster in Fig. 16, despite representing two different machine statuses (specifically, before and after a modification of the water distribution system, as remarked in Section IV-B1). This is due to the fact that the GMM mostly computes "oval" clusters. Its goal, indeed, is to describe the data by means of 2-D Gaussian distributions, which are actually characterized by oval contour lines.

As k-Means, GMM is not able to automatically identify the outliers in a dataset. In the literature, one can find methods that require post-processing clustering results. In [22], the "three times standard deviation principle" is applied to each computed Gaussian component with the aim of identifying each cluster's outliers. Outlier detection algorithms can also be used in combination with the GMM to obtain improved results, as will be discussed in Section VI-A.

In addition to k-Means and GMM, we also consider PCA, which is fundamental for multivariate methods, such as multivariate statistical process control (MSPC) [23]. PCA is mainly adopted for dimensionality reduction and high-dimensional data visualization. For this reason, it can be applied to support clustering. In order to be able to apply PCA to our problem, we use as input data all the statistical features introduced in Table II computed for each CIP cycle and represented as functions of time. These features are properly smoothed, according to (1), and the z-score normalized. In Fig. 17, the PCA-based computed clusters are shown. Only the first two principal components (PCs) are taken into account because they explain more than 90% of the total variability in the dataset. It can be observed that PCA detects five clusters that are mostly formed by consecutive cycles, as represented by their colors. However, the time evolution of the health status of a machine component cannot be correctly identified since there is no intuitive pattern followed by the clusters; along time, that can be inferred from Fig. 17. Therefore, a PCA-based approach turns out not to be appropriate for our problem: this is due to the fact that the features lose their physical meanings because of their transformations in PCs.

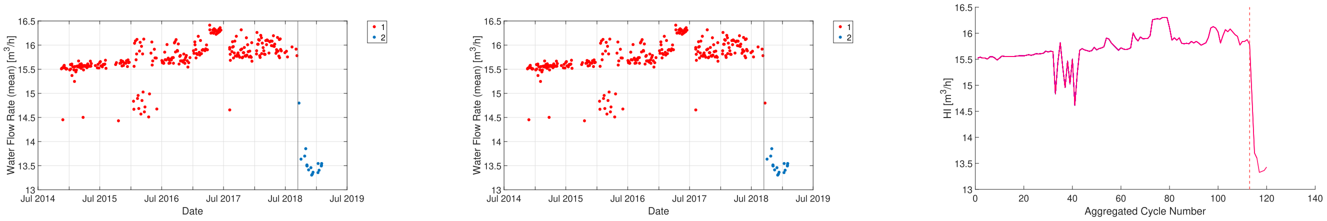
2) Real-Time Monitoring and Application to Predictive Maintenance: As mentioned above, in order to use k-Means and GMM for real-time monitoring, we set the number of clusters to be identified to two. As a matter of fact, two clusters are sufficient to detect an anomalous variation of the health status—should there be no anomalous variations, all the available data would be clustered together.

In order to verify the validity of this approach, in Figs. 18–20, the data clustering results with real-time



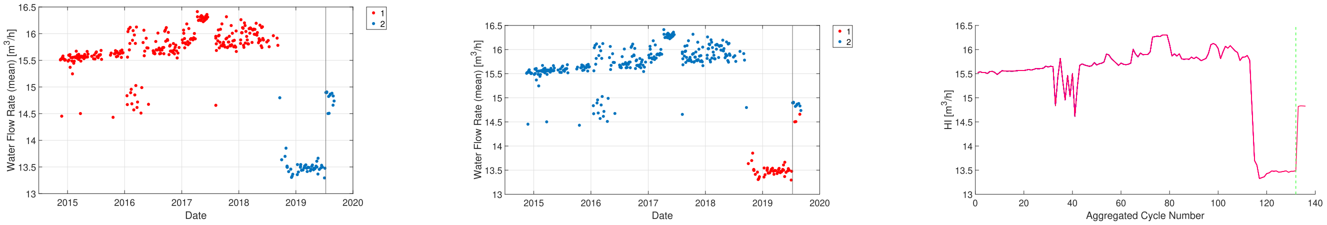
(a) (b) (c)

Fig. 18. Real-time monitoring up to CIP cycle 190: (a) k-Means-based clustering ($k = 2$), (b) GMM-based clustering (number of clusters set a priori to 2), and (c) HI (for both k-Means-based and GMM-based).



(a) (b) (c)

Fig. 19. Real-time monitoring up to CIP cycle 300: (a) k-Means-based clustering ($k = 2$), (b) GMM-based clustering (number of clusters set a priori to 2), and (c) HI (for both k-Means-based and GMM-based).



(a) (b) (c)

Fig. 20. Real-time monitoring up to CIP cycle 340: (a) k-Means-based clustering ($k = 2$), (b) GMM-based clustering (number of clusters set a priori to 2), and (c) HI (for both k-Means-based and GMM-based).

monitoring using (a) k-Means and (b) GMM are shown up to CIP cycles 190, 300, and 340, respectively. In the same figures, (c) corresponding HI is also shown. It can be observed that, in Figs. 18 and 20, neither of the two algorithms can identify the status variations related to the pump maintenance acts (highlighted by vertical lines), unlike DBSCAN in Figs. 11(a) and 13(a). However, both k-Means and GMM succeed in identifying the second cluster after the spray tube leak weld [in Fig. 19(a) and (b)], but it is likely that this is simply due to the large distance of these new cycles from the previous clustered cycles and not to a detected variation of the status.

In terms of real-time HI derivation, k-Means and GMM return the same results since no outlier is identified and then removed. The HIs up to cycles 190, 300, and 340 are shown in Figs. 18(c), 19(c), and 20(c), respectively. Abrupt oscillations can be observed in correspondence to the anomalous behaviors of the water flow rate signals at the beginning of the year 2016. Moreover, although the variations related to the pump maintenance acts are not identified by clustering, it can be noticed that the HI starts deviating at these points.

Overall, it can be concluded that, unlike DBSCAN, k-Means and GMM do not allow tracking the system health status variation by means of both a (time) clustering of sensed data and evaluation of the HI.

D. DBSCAN Versus One-Class SVM and LOF

In order to further validate the use of DBSCAN for outlier identification, we compare its performance with those of two outlier detection algorithms available in the literature, namely, one-class SVM and LOF.

One-class SVM is an unsupervised machine learning technique frequently used to identify the outliers in a dataset. As a matter of fact, it separates the considered faulty samples from the remaining ones by computing a “boundary” around the correct operational data and, consequently, isolating the outliers [24]. The so-called contamination fraction (CF) parameter, corresponding to the outliers’ percentage to be identified, must be set a priori.

LOF is a density-based outlier detection algorithm, as introduced in [25]. It computes the density of each data point and compares it with the density of the neighbors. It identifies the isolated points as outliers. As for one-class SVM, the percentage of the outliers to be detected must be set a priori by the user.

Therefore, since these algorithms are intended only for outlier detection, it is not possible to automatically segment

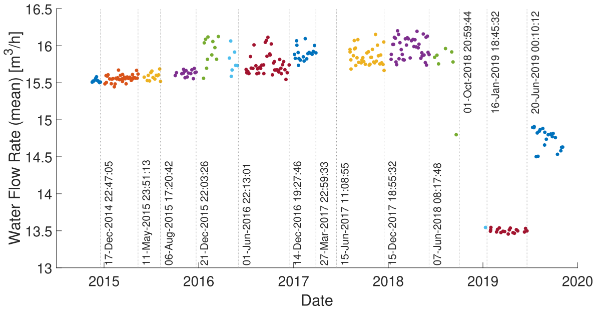


Fig. 21. CIP process: "manual" clustering (time threshold = 300 h) after one-class SVM (CF = 0.2) application.

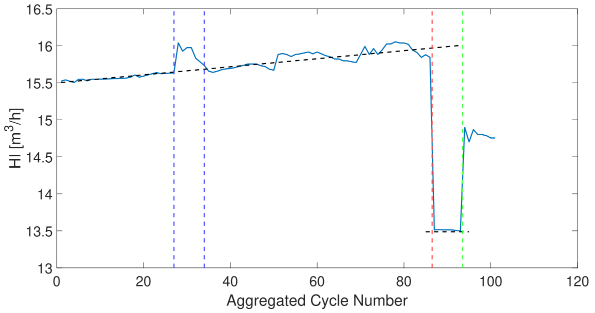


Fig. 22. HI of the condenser water distribution system obtained with one-class SVM (CF = 0.2).

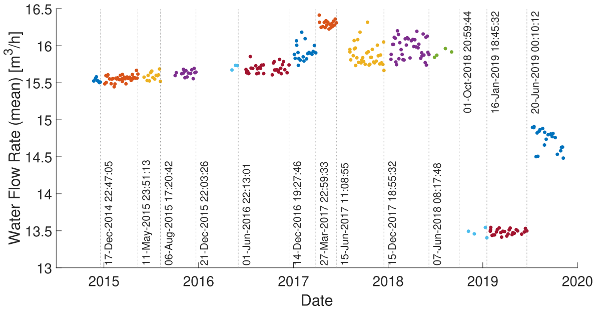


Fig. 23. CIP process: "manual" clustering (time threshold = 300 h) after LOF (CF = 0.2) application.

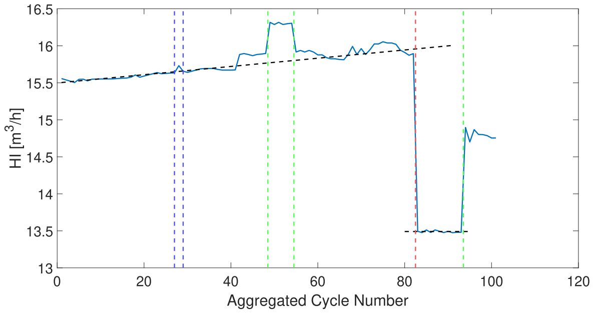


Fig. 24. HI of the condenser water distribution system obtained with LOF (CF = 0.2).

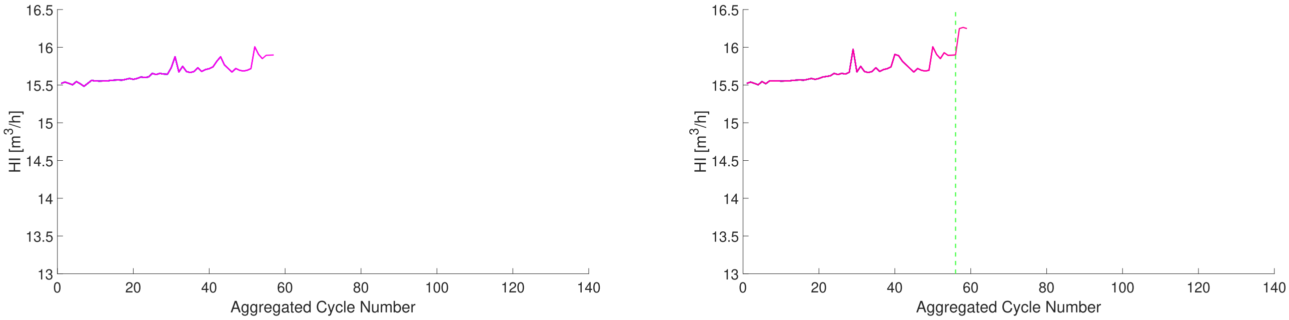
the sensed data (namely, cluster them) and carry out real-time monitoring of the evolution of the considered component health status.

1) A Posteriori Analysis: As anticipated above, since the one-class SVM and LOF provide only outlier detection, no cluster is automatically identified. Therefore, we resort to a "manual clustering," carried out after removing the detected outliers.

The clusters obtained with one-class SVM are shown in Fig. 21. The CF is set to 0.2, i.e., 20% of the data are considered outliers. It can be observed that, unlike Fig. 9, many points belonging to the anomalous clusters (between December 21, 2015, and June 1, 2016) are still present; the cluster included between the two first pump maintenance acts—namely, cluster 4 in Fig. 8—has completely disappeared. The obtained HI, as shown in Fig. 22, is different from the one in Fig. 10. In Fig. 22, two vertical blue lines are inserted to highlight the change in the plot referring to the anomalous clusters considered in the HI computation. Overall, it can be concluded that one-class SVM does not work properly in our case since it removes data points, which would actually help describe the health status of the analyzed machinery, and does not allow identifying all the anomalies, despite the high outliers' percentage indicated by CF.

In Fig. 23, we show the results obtained with LOF, after "manual clustering," with CF set to 0.2, as for the one-class SVM. It can be observed that the anomalous clusters are mostly removed, and the cluster included between the first two pump maintenance acts is still present. As a matter of fact, the corresponding HI, as shown in Fig. 24, is more similar to the one computed with DBSCAN in Fig. 10 than to the HI derived after the one-class SVM outliers' removal shown in Fig. 22. It can be noticed that the change in the plot, highlighted by the two vertical blue lines, is minimum with respect to the one in Fig. 22. Moreover, one can detect the HI deviations due to the first two pump maintenance acts and identify them by two vertical green lines, which are highlighted in Fig. 10, but not in Fig. 22. However, despite the improvements with respect to one-class SVM, it can be concluded that LOF is still not suitable for our purposes because the percentage of outliers to be identified has to be set a priori by the user, and no clustering is carried out automatically.

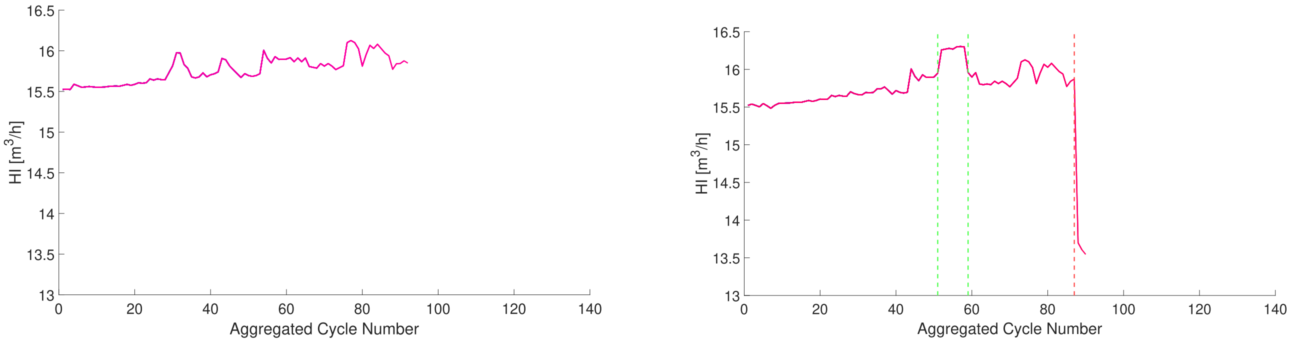
2) Real-Time Monitoring and Application to Predictive Maintenance: As mentioned above, no real-time monitoring of the health status of the components of the condenser water distribution system is possible with both one-class SVM and LOF since these algorithms are not intended for data clustering. However, the HI can still be evaluated in real time. The obtained results are shown in Figs. 25–27 with both (a) one-class SVM and (b) LOF up to CIP cycles 190, 300, and 340, respectively. As in the case of the a posteriori analysis in Section IV-D1, it can be observed that one-class SVM detects as outliers the cycles included between the first two pump maintenance acts: in fact, the HI is quite smooth in this region, as can be seen in Figs. 25(a) and 26(a). On the opposite, the HI obtained with LOF describes the first two maintenance acts on the pump, as can be seen in Figs. 25(b) and 26(b). In Fig. 26(b), it can also be noticed that the HI drops in correspondence to the spray tube maintenance act. On the other hand, in Fig. 27(a), it can be noticed that the HI obtained with one-class SVM correctly captures the variation due to the third pump



(a)

(b)

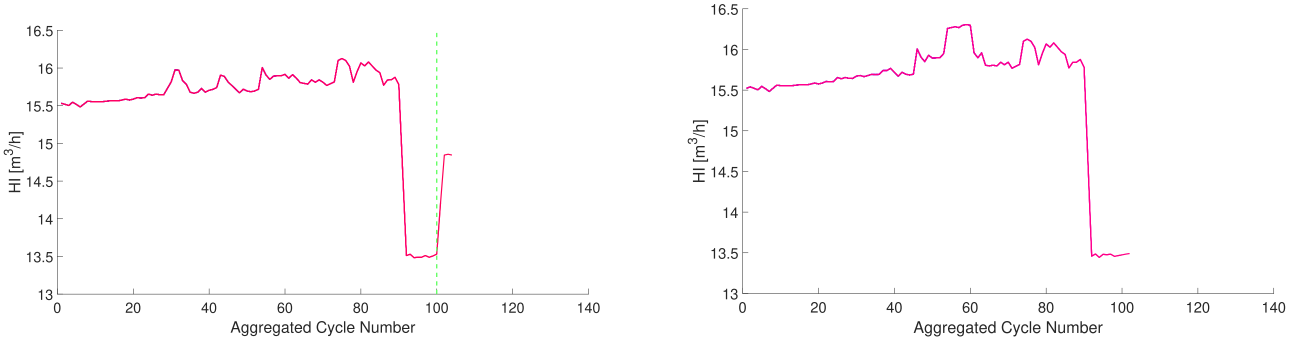
Fig. 25. Real-time HI up to CIP cycle 190: (a) one-class SVM (CF = 0.2) and (b) LOF (CF = 0.2).



(a)

(b)

Fig. 26. Real-time HI up to CIP cycle 300: (a) one-class SVM (CF = 0.2) and (b) LOF (CF = 0.2).



(a)

(b)

Fig. 27. Real-time HI up to CIP cycle 340: (a) one-class SVM (CF = 0.2) and (b) LOF (CF = 0.2).

maintenance act (modification of the water distribution system), unlike the HI obtained with LOF in Fig. 27(b).

Overall, it can be concluded that neither one-class SVM nor LOF allows accurate real-time monitoring.

V. Leak Test

In this section, the proposed data analysis approach is applied to the sensor signals recorded during the LT process. In Section V-A, the single sensor signal processing and the features' monotonicity computation are described. In Section V-B, the procedure for DBSCAN-based a posteriori analysis and real-time monitoring is illustrated. In Section V-C, k-Means is applied as an alternative to DBSCAN, and its performance is compared with that of DBSCAN.

A. Single Sensor Signal Processing and Features' Monotonicity

In order to analyze the LT process and, in particular, the time evolution of the health status of the components contributing to the sealing of the freeze dryer, we consider the signal extracted by the pressure sensor, as mentioned in Section II-B. The procedures considered for pressure signal processing and features' monotonicity computation are illustrated in Sections III-A and III-B, respectively. Eventually, the only feature taken into account is the mean value of the pressure signal. In fact, in Section IV-A, it has been observed that the mean value

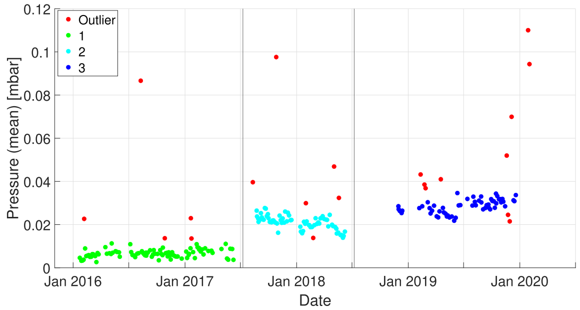


Fig. 28. LT process: DBSCAN-based computed clusters (minPts = 5 and ϵ = 17).

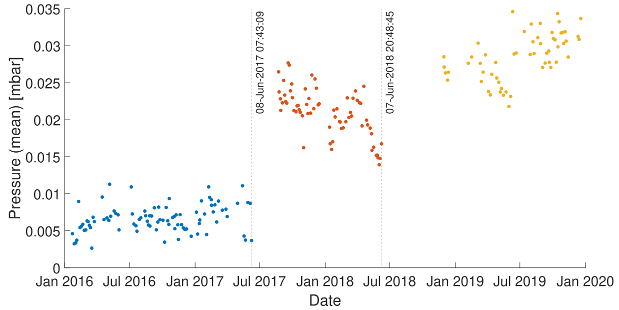


Fig. 29. LT process: "manual" clustering (time threshold = 1500 h).

(of the WFRS signal during "steady-state" conditions) is a feature sufficient to describe the considered process, and the same conclusion applies to LT (based on a similar correlation analysis). We will refer to this feature as "PressureMean." Its monotonicity value, computed following the procedure outlined in Fig. 4 and according to (1) and (2), turns out to be 0.5 (the results are not shown here for the sake of conciseness). This monotonicity value justifies the use of the PressureMean feature for health status monitoring.

B. DBSCAN-Based Analysis and Monitoring

At this point, as discussed in Section III-C, DBSCAN is applied to the two following relevant features:

1) PressureMean;
2) the LT process cycle number (CycleNumber).

The "heuristic" transformation described in Section III-C makes the value of PressureMean numerically comparable to that of CycleNumber.

1) A Posteriori Analysis: For an a posteriori analysis of the sensed data, we follow the approach discussed in Section III-E1.

In the LT case, a good machine status identification is obtained setting minPts = 5 and $\epsilon = 17$: the corresponding clustered data are shown in Fig. 28. The chosen (minPts, ϵ) configuration is the same that allows obtaining, for the CIP process, the data clusters shown in Fig. 8. From the results in Fig. 28, it can be noticed that clusters' separations correspond to repair activities carried out on the analyzed freeze dryer and, consequently, to changes in the machine's operational conditions.

At this point, we follow the HI derivation procedure described in Section III-D, after a "manual clustering" preparatory step. The time threshold is now set to 1500 h. The obtained "manual" clusters are shown in Fig. 29. Unlike what was observed in Section IV-B1, all the identified interruptions are maintenance acts carried out on the freeze-dryer system under analysis. The resulting HI is shown in Fig. 30. Three time intervals, associated with the evolution of the status of the machine, can be clearly identified: during each of these intervals, the HI remains relatively constant. In correspondence to the separation instants between adjacent intervals, the machine was subject to changes, which led to degradation (higher HI). The causes of this counterintuitive phenomenon will be discussed in Section VI-B.

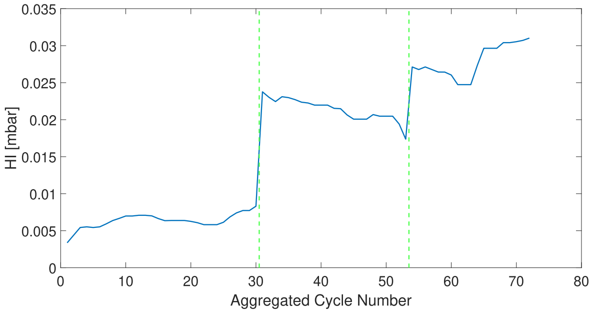


Fig. 30. HI that describes the deterioration of the multiple components that contribute to the sealing of the machine.

2) Real-Time Monitoring and Application to Predictive Maintenance: For the purpose of monitoring in real-time the health status of the multiple components contributing to the sealing of the freeze dryer, the approach discussed in Section III-E2 is followed, as in the CIP case. For illustrative purposes, we will show only the results up to cycles 110 and 180 (the results are updated every ten process cycles), immediately after the two repair activities carried out on the freeze dryer.

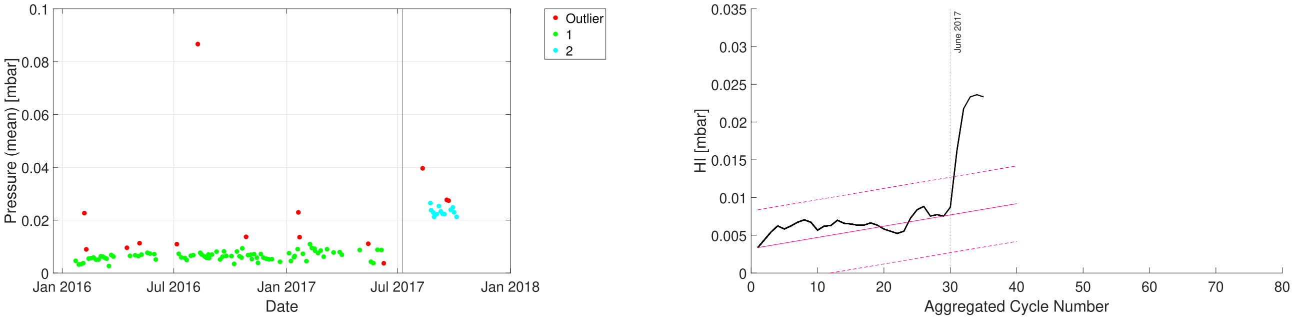
As for DBSCAN-based clustering, the same (minPts, ϵ) configuration used for the CIP process and the LT a posteriori analysis is adopted, namely, minPts $= 5$ and $\epsilon = 17$. The results obtained up to cycles 110 and 180 are shown in Figs. 31(a) and 32(a), respectively. It can be observed that the changes in health status are accurately identified also in real time.

As for the real-time derivation of the HI, the results obtained up to cycles 110 and 180 are shown in Figs. 31(b) and 32(b), respectively. The alarm thresholds are set considering $\Delta = 0.005$ mbar in (5). It can be observed that the HI abruptly increases and, consequently, overcomes the predicted threshold in correspondence to both the status variations (June 2017 and June 2018). Therefore, our predictive approach is efficient also in the LT case.

From the results in Figs. 31(b) and 32(b), it may seem counterintuitive that the HI increases after the maintenance activities. However, this phenomenon will be discussed in Section VI-B.

C. DBSCAN Versus k-Means

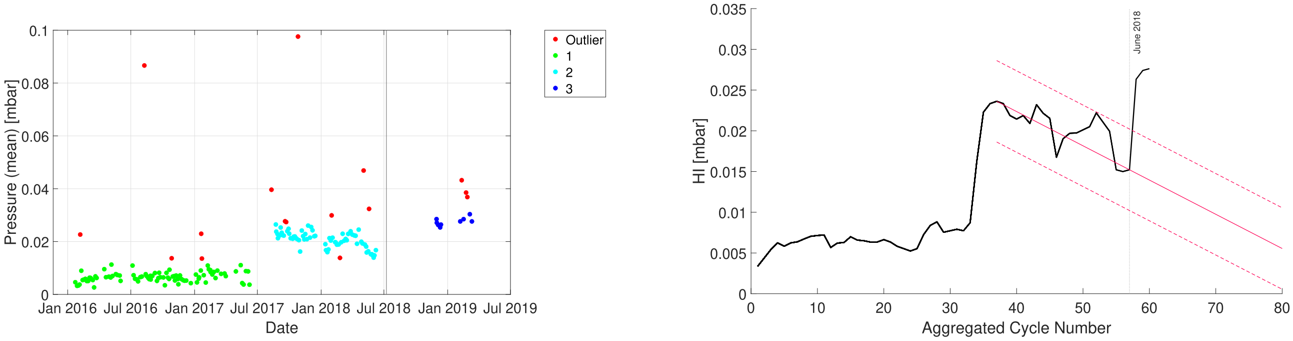
As considered in Sections IV-C and IV-D, the performance of DBSCAN has been compared with those of other clustering and outlier detection algorithms. For the sake of conciseness, we show only the results obtained with k-Means.



(a)

(b)

Fig. 31. Real-time monitoring up to LT cycle 110: (a) DBSCAN-based clustering (minPts = 5 and $\epsilon = 17$) and (b) HI with linear interpolation and alarm thresholds.



(a)

(b)

Fig. 32. Real-time monitoring up to LT cycle 180: (a) DBSCAN-based clustering (minPts = 5 and $\epsilon = 17$) and (b) HI with linear interpolation and alarm thresholds.

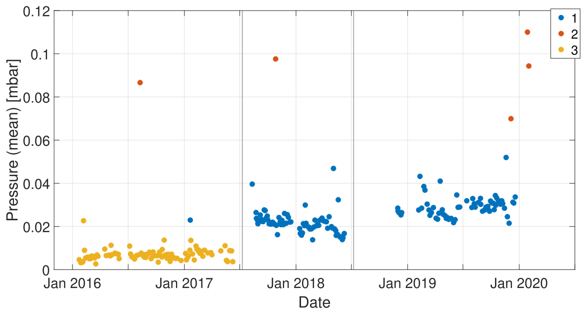


Fig. 33. LT process: k-Means-based computed clusters ($k = 3$).

1) A Posteriori Analysis: For a fair comparison with DBSCAN, in the k-Means case, k is set to 3. In Fig. 33, it can be observed that the three clusters do not comply with the maintenance acts. Therefore, as in the CIP case, k-Means does not perform effectively in the LT case either.

2) Real-Time Monitoring and Application to Predictive Maintenance: As discussed in Section IV-C2, k-Means requires that the user sets a priori the number of clusters to be identified. As previously considered, we set a priori the number of clusters to 2 in order to highlight an anomalous variation of the health status of the multiple components involved in the LT process. The obtained results up to cycles 110 and 180 are shown in Figs. 34(a) and 35(a), respectively. In the former case, it can be observed that k-Means cannot detect the changes in the status since some cycles before the considered repair activity belong to the newly formed cluster after it. In the latter case, the cycles after the repair activity are included in the same cluster as the cycles before it.

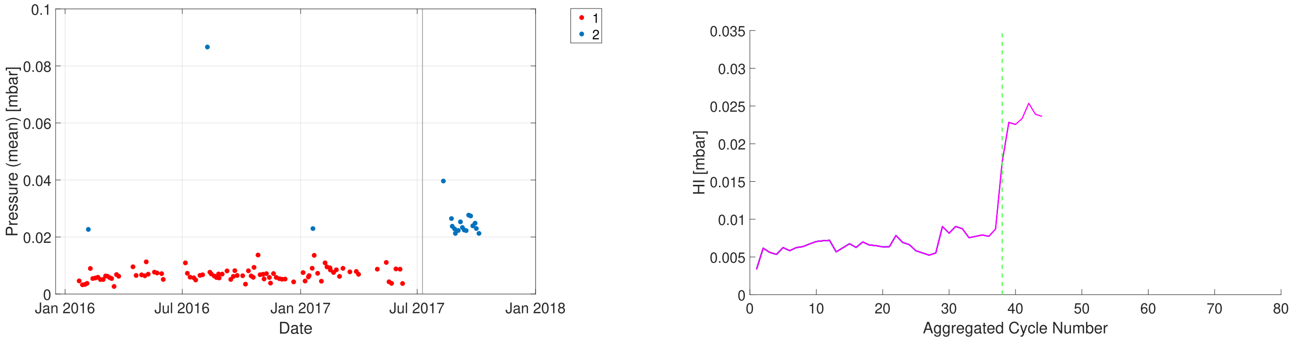
The corresponding HIs, up to cycles 110 and 180, are shown in Figs. 34(b) and 35(b), respectively. By comparing these results with those in Figs. 31(b) and 32(b), it can be observed that the overall behaviors are the same but for some oscillations due to the outliers that, using k-Means instead of DBSCAN, cannot be detected and removed.

VI. Discussion

It is remarkable that the same DBSCAN-based semiautomatic method can be used to describe the evolution of two different processes (namely, CIP and LT), starting from two signals of different natures (namely, water flow rate and pressure). However. for the different natures of the used sensors, the proposed clustering methodology (including its parametric values) is the same. Although similar results, in terms of clustering and outlier detection accuracy, could have been achieved by a trained operator, it is important to emphasize that we succeeded in making our real-time monitoring approach automatic and, therefore, not vulnerable to human errors and not requiring any manual handling.

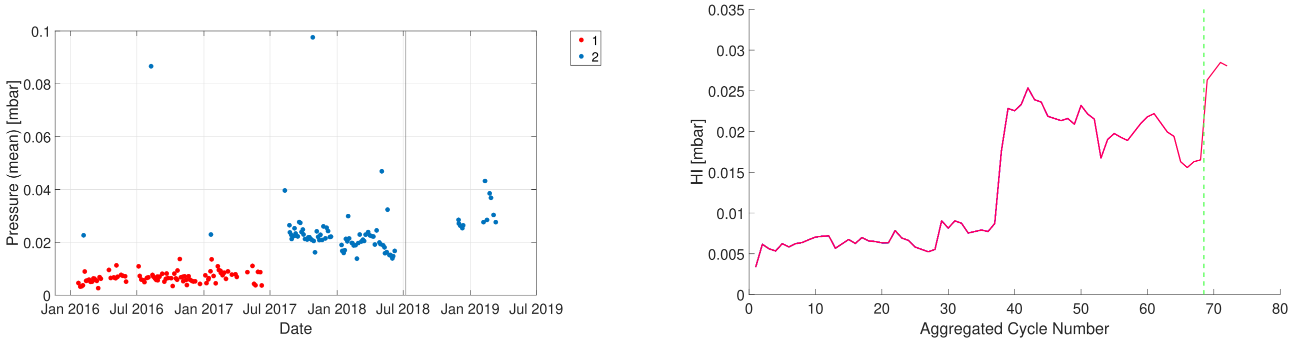
A. Comparison With Other Algorithms

In Section IV-C, we have compared DBSCAN with two different clustering algorithms, namely, k-Means and GMM. In Section IV-D, the comparison was between DBSCAN and two different outlier detection algorithms, namely, one-class



(a) (b)

Fig. 34. Real-time monitoring up to LT cycle 110: (a) *k*-Means-based clustering ($k = 2$) and (b) HI.



(a) (b)

Fig. 35. Real-time monitoring up to LT cycle 180: (a) *k*-Means-based clustering ($k = 2$) and (b) HI.

SVM and LOF. In Section V-C, DBSCAN has been compared with *k*-Means. In all cases, both a posteriori analysis and real-time monitoring (with application to predictive maintenance) have been considered.

Monitoring of time evolution of the considered component health status is possible using all the considered algorithms, namely, *k*-Means, GMM, one-class SVM, and LOF. This monitoring can be performed by means of the real-time HI evaluation (provided that the unremoved outliers do not impair the evaluation). However, our goal is also to track the variations in the health status that led to the changes detected through the HI by means of clustering. Therefore, it can be concluded that, for both considered processes (CIP and LT), the most efficient algorithm is DBSCAN because it can simultaneously cluster the machine statuses and the outliers.

Comparisons with other algorithms, such as AutoEncoder-based approaches, are not feasible since the available sensor data describing both correct operational conditions and anomalous behaviors of the analyzed freeze dryer are not sufficient to train the neural network to accurately detect the system faults.

In order to further validate our approach based on DBSCAN, we combine a clustering algorithm with an outlier detection algorithm. For simplicity, we consider only the CIP process. We choose LOF to remove the outliers, and then, we apply GMM to cluster the remaining data points. The number of components of GMM (set to 7) and CF (set to 0.2) is the same used in Sections IV-C and Section IV-D, respectively. The results are shown in Fig. 36. It can be observed that the correct clusters are not identified. For instance, cluster 1 turns out not to be coherent with the maintenance acts in its corresponding time interval: as a matter of fact, it includes all the CIP cycles between the first two pump maintenance acts (i.e., March 27, 2017, and June 15, 2017) and other cycles belonging to the two adjacent statuses. Therefore, even in this case, the performance of DBSCAN, with reference to the a posteriori analysis, is not reached.

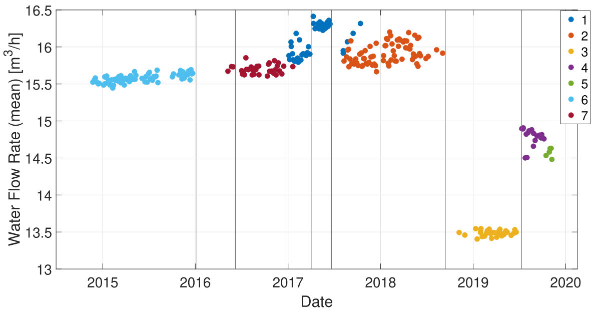


Fig. 36. CIP process: clusters identified with GMM (with seven components) and LOF-based outlier removal (with CF = 0.2).

B. Failure Modes

Although DBSCAN has been shown to be the most efficient clustering algorithm for the problem at hand, it still has some disadvantages. For instance, the clustering results are sensitive to the choice of the two parameters ϵ and minPts. In particular, the chosen value of ϵ is critical because, as mentioned in Section III-C, one has to correctly identify the distance corresponding to the "elbow" formed in the graph representing the minPtsth nearest point distances. Moreover, DBSCAN does not perform well in the presence of high-dimensional data. This might be the case when using more sensor data or when more than one feature turns out to be sufficiently monotonous.

Another limitation of our approach is associated with the computation of the HI. For instance, from the results in Fig. 30, it can be observed that the HI continues to increase even after the two maintenance acts (highlighted in green)—an identical behavior can be observed in Figs. 31(b) and 32(b). This is likely due to phenomena associated with the physics of the considered processes. Therefore, our algorithm could be improved by taking into account the physical characteristics of the freeze dryer and its processes [26].

C. Time Segmentation

As observed in Sections IV-B1 and IV-B2 for CIP and Sections V-B1 and V-B2 for LT, it turns out that, for our purposes, it is sufficient to extract only one feature for both the analyzed processes, namely, the mean of the water flow rate signal at steady-state conditions (for CIP) and the mean of the pressure signal (for LT)—denoted as FlowMean (see Section IV-A) and PressureMean (see Section V-A), respectively. Clustering is then applied to these extracted features together with the process cycle number in order to carry out a time-aware analysis. However, since both the extracted features mentioned above for CIP (FlowMean) and LT (PressureMean) are expressed as functions of time, other time series segmentation approaches (not based on clustering) could be considered.

A segmentation algorithm revolves around a linear approximation of the available time series data [27]. The approximation can be carried out by means of either linear interpolation or linear regression. One way to evaluate the segmentation accuracy is by computing the mean square error (mse) between the actual time series data points and their approximated values. Another way to evaluate the approximation error is by calculating the L_∞ norm between the approximating line and the time series data points. Regardless of the chosen error metric (mse or L_∞ norm), the following two types of approximation error can be considered: segment error and segmentation error. The *segment error* measures the difference between the actual data points and the approximated ones (in the approximating line) for each identified segment. The *segmentation error* provides an estimate of the overall difference between the actual time series and the approximation (given by a concatenation of segments). Therefore, in order to carry out an accurate time series segmentation, both segment error and segmentation error must be below two different user-defined thresholds, denoted as "max error" and "total max error," respectively. In the following, we summarize a few relevant segmentation approaches.

In general, three approaches for time series segmentation can be considered [27]: 1) top-down; 2) bottom-up; and 3) sliding window. A *top-down approach* consists of recursively partitioning the time series until a stopping criterion is met (namely, all the segments have approximation errors below "max error" or the desired number of segments is reached). A *bottom-up approach* foresees, first, the subdivision of the considered time series into a large number of segments, which are progressively merged, until a stopping criterion (namely, the segmentation error becomes greater than "total max error" or the desired number of segments is reached) is satisfied.

Top-down and bottom-up are "off-line" approaches since they require analyzing the entire dataset at once. This makes these two approaches relevant for a posteriori analysis but not for real-time monitoring and predictive maintenance. On the opposite, a *sliding window approach* is an online approach. This approach starts from the first point of the analyzed time series and creates a segment of increasing length until the associated segment error becomes greater than the "max error." At this point, another segment of increasing length starts being created from the data point next to the end of the last identified segment. As mentioned above, the time series to be segmented are FlowMean (for CIP) and PressureMean (for LT). The same linear interpolation used for the HI predictive model can be applied to approximate the time series. A change in the machine health status can be identified when the approximation error between FlowMean/PressureMean and their interpolating line becomes greater than the "max error"; this leads to the identification of a new segment. The selection of the "max error" value is, thus, critical. As a matter of fact, this threshold value should depend on the data itself, and therefore, it would be complicated to set it a priori, especially for real-time monitoring when the entire dataset is not available at once. Selecting an appropriate value for the approximation error threshold is crucial because it affects the accuracy of the identification of the system health status: if the threshold is too small, many health statuses may be erroneously detected. On the opposite, if the threshold is too large, it can happen that critical health status changes (representative of significant variations of the machine's operational conditions) are not correctly identified.

Another approach for time series segmentation relies on the use of a hidden Markov model (HMM). An HMM is associated with a pair of stochastic processes, namely, a "hidden" process and an observable process [28]. The *hidden process* is Markovian (i.e., the probability that the process is in a given state at a given epoch depends only on the state visited at the previous epoch) and can assume a finite number of values. At every epoch, this process can visit another state or remain in the same state. The *observable process*, at every time epoch, generates a sample from a normal distribution with a mean value depending on the current state. An analyzed time series can be seen as a realization of the observable process. A segment is defined as the time interval over which the hidden process remains in the same state. Therefore, segmenting the time series is equivalent to estimating the underlying state sequence of the hidden process. In our case, the observable process, as a function of the cycle number, would be either FlowMean (for CIP) or PressureMean (forLT). On the other hand, the underlying state sequence to be estimated would correspond to the evolution of the health status of the considered machine. The difficulties arise when the parameters characterizing the HMM (e.g., the number of states to be identified, the transition probability matrix of the hidden process, the mean and the standard deviation of the conditionally independent random variables forming the observable process) need to be estimated in a so-called "parameter estimation step," which requires extra processing.

In general, the applicability of time series segmentation approaches to our problem, in a comparative way with respect to the proposed DBSCAN-based method, is an interesting research direction.

D. Multidimensional Extension

As mentioned in Section III-D, should more than one feature turn out to be sufficiently monotonous for the HI derivation, an extension to a multidimensional approach would be required. In this case, one should "fuse" multiple features in order to obtain a unique indicator for the health status of the considered machine component. This could be achieved by means of PCA or using neural networks, as proposed in prognostic literature [29], and is the subject of current research activity.

Further multidimensional extensions rely on the use of multivariate time series. However, this goes beyond the scope of this work since our focus is on the performances that can be achieved by using the data extracted from a single sensor. Multivariate time series will be of significantly higher interest in the presence of multiple sensors.

VII. Conclusion

In this work, a time-aware clustering approach for the computation of an HI of system components of an industrial pharmaceutical machine (namely, a freeze dryer) has been proposed. It has been tested with two different signals (water flow rate and pressure) acquired during two different processes, namely, CIP and LT. Our results show that an accurate identification of the evolution of the different health conditions of the considered system can be obtained by means of a time-aware DBSCAN-based clustering for both a posteriori analysis and real-time monitoring. In the context of real-time monitoring, a predictive maintenance approach (based on linear interpolation) has been proposed, verifying its efficiency in both the CIP and LT cases. A comparison with other clustering algorithms (namely, k-Means and GMM) and outlier detection algorithms (namely, one-class SVM and LOF) has been carried out, highlighting the superiority of DBSCAN. A qualitative comparison with other (nonclustering-based) time series segmentation approaches (in particular, online methods such as sliding windows and HMM) has also been provided. Future research activities will focus on the application of the proposed predictive maintenance approach to other industrial machines, possibly using multiple sensors and multiple features.

Acknowledgment

GlaxoSmithKline (GSK) S.p.a. provided the process data used for this article. The authors would like to thank Christine Cahyadi, Micaela Prati, and Damiano Trento, all from GSK (Micaela Prati was with GSK when this work was carried out), for useful discussions and support.

References

[1] J. S. Barrett and S. P. Koprowski, "The epiphany of data warehousing technologies in the pharmaceutical industry," *Int. J. Clin. Pharmacol. Therapeutics*, vol. 40, no. 3, pp. S3–13, Mar. 2002.

[2] X. Z. Wang, *Data Mining and Knowledge Discovery for Process Monitoring and Control*. London, U.K.: Springer, 2012.

[3] J. Chen and K.-C. Liu, "On-line batch process monitoring using dynamic PCA and dynamic PLS models," *Chem. Eng. Sci.*, vol. 57, no. 1, pp. 63–75, 2002, doi: 10.1016/S0009-2509(01)00366-9.

[4] J. MacGregor and A. Cinar, "Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods," *Comput. Chem. Eng.*, vol. 47, pp. 111–120, Dec. 2012, doi: 10.1016/j.compchemeng.2012.06.017.

[5] R. S. Beebe, *Predictive Maintenance of Pumps Using Condition Monitoring*. Amsterdam, The Netherlands: Elsevier, 2004.

[6] R. A. A. Habeeb, F. Nasaruddin, A. Gani, I. A. T. Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: A survey," *Int. J. Inf. Manage.*, vol. 45, pp. 289–307, Apr. 2019, doi: 10.1016/j.ijinfomgt.2018.08.006.

[7] T. P. Carvalho et al., "A systematic literature review of machine learning methods applied to predictive maintenance," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106024, doi: 10.1016/j.cie.2019.106024.

[8] H. Skima, K. Medjaher, and N. Zerhouni, "Accelerated life tests for prognostic and health management of MEMS device," in *Proc. 2nd Eur. Conf. Prognostics Health Manag. Soc. (PHM)*, vol. 2, no. 1, Nantes, France, Jul. 2014, pp. 1–7. [Online]. Available: http://www.papers.phmsociety.org/index.php/phme/article/view/1527

[9] T. Wang, "Trajectory similarity based prediction for remaining useful life estimation," Ph.D. dissertation, Dept. Ind. Eng., Univ. Cincinnati, Cincinnati, OH, USA, Aug. 2010.

[10] G. Calzavara, E. Oliosi, and G. Ferrari, "A time-aware data clustering approach to predictive maintenance of a pharmaceutical industrial plant," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Jeju Island, South Korea, Apr. 2021, pp. 454–458, doi: 10.1109/ICAIIC51459.2021.9415206.

[11] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Portland, OR, USA, Aug. 1996, pp. 226–231. [Online]. Available: https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf

[12] K. Williams. (Mar. 2016). *How to Freeze Dry Faster*. [Online]. Available: https://www.labconco.com/articles/how-to-freeze-dry-faster

[13] J. B. Ali, L. Saidi, S. Harrath, E. Bechhoefer, and M. Benbouzid, "Online automatic diagnosis of wind turbine bearings progressive degradations under real experimental conditions based on unsupervised machine learning," *Appl. Acoust.*, vol. 132, pp. 167–181, Mar. 2018, doi: 10.1016/j.apacoust.2017.11.021.

[14] L. Saidi, J. B. Ali, E. Bechhoefer, and M. Benbouzid, "Wind turbine high-speed shaft bearings health prognosis through a spectral Kurtosis-derived indices and SVR," *Appl. Acoust.*, vol. 120, pp. 1–8, May 2017, doi: 10.1016/j.apacoust.2017.01.005.

[15] A. Mosallam, K. Medjaher, and N. Zerhouni, "Component based data-driven prognostics for complex systems: Methodology and applications," in *Proc. 1st Int. Conf. Rel. Syst. Eng. (ICRSE)*, Beijing, China, Oct. 2015, pp. 1–7, doi: 10.1109/ICRSE.2015.7366504.

[16] J. Coble and J. W. Hines, "Identifying optimal prognostic parameters from data: A genetic algorithms approach," in *Proc. Annu. Conf. Prognostics Health Manage. (PHM) Soc.*, vol. 1, no. 1, San Diego, CA, USA, Sep. 2009, pp. 1–11. [Online]. Available: https://papers.phmsociety.org/index.php/phmconf/article/view/1404

[17] K. Rahul, R. Agrawal, and A. Pal, "Color image quantization scheme using DBSCAN with K-means algorithm," in *Intelligent Computing, Networking, and Informatics* (Advances in Intelligent Systems and Computing), vol. 243. New Delhi, India: Springer, 2014, pp. 1037–1045, doi: 10.1007/978-81-322-1665-0_106.

[18] E. Kreyszig, *Advanced Engineering Mathematics*, 4th ed. New York, NY, USA: Wiley, 1979.

[19] J. Yadav and M. Sharma, "A review of K-mean algorithm," *Int. J. Eng. Trends Technol.*, vol. 4, no. 7, pp. 2972–2976, Jul. 2013. [Online]. Available: http://www.ijettjournal.org/volume-4/issue-7/IJETT-V4I7P139.pdf

[20] J. Anitha, I.-H. Ting, S. A. Agnes, S. I. A. Pandian, and R. V. Belfin, *Systems Simulation and Modeling for Cloud Computing and Big Data Applications*, 1st ed. New York, NY, USA: Academic, 2020.

[21] R. Pamula, J. K. Deka, and S. Nandi, "An outlier detection method based on clustering," in *Proc. 2nd Int. Conf. Emerg. Appl. Inf. Technol.*, Kolkata, India, Feb. 2011, pp. 253–256, doi: 10.1109/EAIT.2011.25.

[22] W. Liu, D. Cui, Z. Peng, and J. Zhong, "Outlier detection algorithm based on Gaussian mixture model," in *Proc. IEEE Int. Conf. Power, Intell. Comput. Syst. (ICPICS)*, Shenyang, China, Jul. 2019, pp. 488–492, doi: 10.1109/ICPICS47731.2019.8942474.

[23] B. M. Wise, N. Ricker, D. Veltkamp, and B. Kowalski, "A theoretical basis for the use of principal component models for monitoring multivariate processes," *Process Control Qual.*, vol. 1, no. 1, pp. 41–51, 1990. [Online]. Available: https://www.semanticscholar.org/paper/A-Theoretical-Basis-for-the-use-of-Principal-Models-Wise-Ricker/7fc07c038b92dab993da65a581d1754bae55d402

[24] S. Mahadevan and S. L. Shah, "Fault detection and diagnosis in process data using one-class support vector machines," *J. Process Control*, vol. 19, no. 10, pp. 1627–1639, 2009, doi: 10.1016/j.jprocont.2009.07.011.

[25] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, Dallas, TX, USA, May 2000, pp. 93–104, doi: 10.1145/335191.335388.

[26] G. Calzavara, L. Consolini, and G. Ferrari, "Leak detection and classification in pharmaceutical freeze-dryers: An identification-based approach," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 1568–1573, doi: 10.1109/CDC45484.2021.9683753.

[27] E. J. Keogh, S. Chu, D. M. Hart, and M. J. Pazzani, "Segmenting time series: A survey and novel approach," in *Data Mining in Time Series Databases*. Singapore: World Scientific, Jun. 2004, pp. 1–21, doi: 10.1142/9789812565402_0001.

[28] A. Kehagias, "A hidden Markov model segmentation procedure for hydrological and environmental time series," *Stochastic Environ. Res. Risk Assessment*, vol. 18, pp. 117–130, Apr. 2004, doi: 10.1007/s00477-003-0145-5.

[29] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, May 2017, doi: 10.1016/j.neucom.2017.02.045.

Eleonora Oliosi received the bachelor's degree (three-year program) in information, electronic, and telecommunication engineering from the University of Parma, Parma, Italy, in March 2018, and the master's degree (second cycle degree) in communication engineering from the University of Parma in October 2020, with a thesis titled *Statistical Analysis of Sensor Data for Predictive Maintenance of Industrial Lyophilizers*. She is currently pursuing the Ph.D. degree in automotive engineering for intelligent mobility with the University of Bologna, Bologna, Italy.

She is with the Internet of Things (IoT) Laboratory, University of Parma. Her research interests also revolve around 5G and autonomous driving.

Gabriele Calzavara received the master's degree in mathematics and the Ph.D. degree in information engineering from the University of Parma, Parma, Italy, in 2018 and 2022, respectively.

During his Ph.D. degree, he worked on developing data-driven techniques for identifying and detecting faults in industrial monitoring, with a specific application to a pharmaceutical freeze dryer. After completing his Ph.D. degree, he started working as a Data Scientist at Ammagamma S.r.l., Modena, Italy, where he applies his data analysis and modeling skills to provide effective solutions to business challenges.

Gianluigi Ferrari (Senior Member, IEEE) received the Ph.D. degree in information technologies from the University of Parma, Parma, Italy, in 2002.

He is a Full Professor of Telecommunications with the University of Parma, where he has been a Coordinator of the Internet of Things (IoT) Laboratory Since September 2006. Since 2016, he has been the Co-Founder and the President of things2i S.r.l., Parma, a spin-off company of the University of Parma dedicated to IoT and smart systems. He is currently the Head of the bachelor's degree in computer engineering, electronics, and telecommunications at the University of Parma. He has published and consulted extensively in these areas, coordinating several technical projects, including EU-funded competitive projects. His research activities revolve around signal processing, communication/networking, and IoT.

Dr. Ferrari is a member of the Scientific Council of the INSIDE Industry Association.