Enhancing manufacturing intelligence through an unsupervised data-driven methodology for cyclic industrial processes

Journal Pre-proof

Enhancing manufacturing intelligence through an unsupervised data-driven methodology for cyclic industrial processes

Tania Cerquitelli, Francesco Ventura, Daniele Apiletti, Elena Baralis, Enrico Macii, Massimo Poncino

PII: S0957-4174(21)00700-4
DOI: https://doi.org/10.1016/j.eswa.2021.115269
Reference: ESWA 115269

To appear in: *Expert Systems With Applications*

Received date : 13 December 2019
Revised date : 19 March 2021
Accepted date : 20 May 2021

Please cite this article as: T. Cerquitelli, F. Ventura, D. Apiletti et al., Enhancing manufacturing intelligence through an unsupervised data-driven methodology for cyclic industrial processes. *Expert Systems With Applications* (2021), doi: https://doi.org/10.1016/j.eswa.2021.115269.

Enhancing manufacturing intelligence through an unsupervised data-driven methodology for cyclic industrial processes

Tania Cerquitelli[a], Francesco Ventura[a], Daniele Apiletti[a,*], Elena Baralis[a], Enrico Macii[b], Massimo Poncino[a]

[a] *Department of Control and Computer Engineering, Politecnico di Torino, corso Duca degli Abruzzi, 24, Torino, Italy*
[b] *Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, corso Duca degli Abruzzi, 24, Torino, Italy*

Abstract

Recent trends in intelligent manufacturing are transforming shop floor environments into digital factories, thanks to a pervasive integration of information and communication technologies in production lines. Industrial processes become the source of high-volume heterogeneous data, paving the way to create manufacturing intelligence by means of machine learning and data-driven methodologies. In such settings, predictive diagnostics play a crucial role, as they promise to predict future critical conditions in the production process. Unfortunately, the diffusion of data-driven predictive maintenance methodologies is limited by (i) the absence of timely ground-truth knowledge (i.e., class labels), required in the learning phase of data-driven supervised approaches, and (ii) the limited availability of data-mining expertise among application-domain experts, required to harness the power of machine learning techniques. Innovative data-driven services are needed to support domain experts in (i) applying powerful self-learning intelligent techniques with limited technical expertise and (ii) easily understanding results and choices operated by such in-

*Corresponding author, phone: +39-011-090-7084, Politecnico di Torino, Italy
Email addresses: tania.cerquitelli@polito.it (Tania Cerquitelli), francesco.ventura@polito.it (Francesco Ventura), daniele.apiletti@polito.it (Daniele Apiletti), elena.baralis@polito.it (Elena Baralis), enrico.macii@polito.it (Enrico Macii), massimo.poncino@polito.it (Massimo Poncino)

telligent techniques, to increase trust by means of transparency. To this aim, this paper presents UDaMP, an integrated platform to support manufacturing intelligence by providing a transparent, self-tuning, unsupervised discovery and assisted data labelling service for predictive maintenance, specifically targeted at cyclic industrial processes. UDaMP includes (i) production-cycle-aware feature engineering, (ii) unsupervised discovery of production-cycle categories, (iii) self-tuning of the optimal number of categories, (iv) human-readable characterisation of production-cycle categories, and (v) assisted data labelling for domain experts. Scalable clustering algorithms automatically discover groups of production cycles sharing common time-independent properties. A self-tuning strategy is integrated to automatically configure the specific input parameter and select the best approach for the data under analysis. Each cluster is then locally characterised through the data distribution of the top 10 most relevant features to support domain experts in uncovering its meaning. Experimental evaluation of UDaMP has been performed on real-world data collected in two different industrial settings.

Keywords: Cluster analysis, Self-tuning machine learning, Industry 4.0, Predictive maintenance, Data Analytics.

1. Introduction

The advent of the fourth industrial revolution (Industry 4.0) has led to Manufacturing Intelligence (Davis et al., 2012; Chien et al., 2013; Chen & Chien, 2011; Chien et al., 2010) thanks to a fully-integrated production environment with Internet of Things. This new revolutionary model allows a continuous monitoring of production processes, thus generating high volumes of data collected in real-time. Data-driven methodologies can bring to the surface a rich spectrum of valuable items in the form of knowledge useful for adding intelligence in production and industrial plants. For example, production managers are able to better assess, analyse and therefore understand shop floor activities, and sometimes directly translate the extracted knowledge into actions. In this

scenario the need for effective and efficient architectures capable of processing large volumes of data and analysing them effectively is becoming increasingly important. Predictive maintenance approaches represent a key asset to significantly reduce maintenance costs and improve productivity. Unfortunately, the actual exploitation of the aforementioned data-driven approaches are limited by the absence of standard ground-truth knowledge (i.e., class labels) on equipment conditions and their characterisation during use. Innovative unsupervised data-driven methods are needed to help the domain experts in the definition of the production cycle classification.

This paper presents UDaMP (Unsupervised Data-driven Methodology for Production-cycles Characterisation) an integrated platform to provide self-tuning semi-supervised data labelling to make predictive maintenance more valuable in manufacturing even when ground truth is not available to train a model.

UDaMP has been tested on two different real-world industrial use cases: a robotic industry and a white goods company, where it supported domain experts in discovering homogeneous groups of production cycles, understanding their meaning, and consequently labelling production-cycle groups without standard ground-truth knowledge (i.e., class labels).

The main research contributions of this work can be summarised in the following:

- The introduction of an automatic feature-engineering strategy tailored to cycle-based signals coming from industrial production processes.
- A new strategy to characterise slowly-degrading industrial processes over time by aggregating intra-cycle features.
- A new semi-supervised data labelling process exploiting the evaluation of a computationally-scalable silhouette-based unsupervised index.
- A new interpretable strategy for characterising the unsupervised partitioning process by exploiting an interpretable CART model and descriptive statistics of the most relevant features.

This paper is organised as follows. Section 2 discusses existing literature in manufacturing intelligence. Section 3 presents the main building blocks of the proposed approach tailored to the Industry 4.0 and describes the proposed data-driven methodology to support the domain expert in labelling production cycles without ground-truth knowledge (i.e., class labels) on equipment conditions. Section 4 presents the technological architecture of UDaMP, and Section 5 reports the different case studies on which UDaMP has been tested and their experimental results. Section 6 presents a discussion on experimental results, impact and open issues related with the proposed approach and its applications. Finally, section 7 draws conclusions and presents future research directions.

2. Related works

With the advent of Industry 4.0, companies are able to increase productivity and reliability by controlling and predicting maintenance interventions through state-of-the-art smart systems. The adoption of Data Integration, Data Management, and Data-Driven algorithms in manufacturing environments has been recently recognised to belong to the concept of Manufacturing Intelligence, both in industry and in the scientific literature (Davis et al., 2012; Chien et al., 2013; Chen & Chien, 2011; Chien et al., 2010). Indeed, it has been highlighted that modern data management and analytics pipelines applied to manufacturing-related data can be used to extract valuable knowledge and derive decision rules, that can enhance production efficiency and effectiveness, bringing benefit to business decisions as well.

In (Wang et al., 2016), the authors address the modern industrial scenario which necessitates flexible tools and platforms to process great amounts of data collected by production methods, by presenting a smart factory framework. To this aim, industrial IoT (Internet of Things) networks, cloud platforms, and supervisory control terminals including smart machines, conveyors and products are integrated and exploited. The result is a totally self-organised system managing feedback data and coordinating the central control system to achieve high

efficiency. In (Lee et al., 2014), the trend of manufacturing transformations in industry 4.0 environments is discussed. The paper analyses the level of preparation of IT tools in managing industrial Big Data and predicting maintenance operations. In (Yan et al., 2017), a framework for structuring multi-source contrasted data is proposed, contemplating spatio-temporal properties and modelling invisible factors. It is a step toward a totally transparent production process, which would allow prompt implementation of predictive maintenance and provide remaining-life predictions of key components of machine apparatus.

As depicted in (Marques et al., 2019) many modern scenarios rely on complex IoT platforms enabling new opportunities along with new challenges. In (D'silva et al., 2017) and (Apiletti et al., 2018) examples of real-time data processing architectures are illustrated. Both papers present distributed architectures based on open source state-of-the-art frameworks (i.e. Apache Kafka, Apache Spark, and Apache Cassandra) which ensure reliability and scalability for IoT sensor networks. While (D'silva et al., 2017) include an integrated visualisation tool, (Apiletti et al., 2018) provide a self-tuning engine for predictive maintenance able to define possible equipment failure and intervention needs.

A multitude of attempts to cut down the necessity of domain experts and lower the running costs of machine learning algorithms have been described in (Ribeiro et al., 2015; Yao et al., 2017). Feasible machine-learning solutions are based on an architecture able to create a flexible and scalable data-driven service, as proposed in (Ribeiro et al., 2015). It exploits an open-source solution with real-world sensors and weather data, to analyse predictions of electricity demand. In (Yao et al., 2017), an empirical comparison of MLaaS (Machine Learning as a Service) platforms is presented. The authors evaluated the potential of fully-automated systems, turnkey systems and fully-customizable systems.

At last, unsupervised algorithms are powerful methodologies that found applications in a wide range of scenarios besides Industry 4.0 such as in (Park et al., 2019) and in (Cerquitelli et al., 2018), and they are continuously evolving with new approaches (Barak & Mokfi, 2019; Ünlü & Xanthopoulos, 2019). In

(Park et al., 2019), the authors propose an advanced clustering approach tailored to textual data to exploit the syntactically and semantically meaningful features that can be extracted from documents. In (Cerquitelli et al., 2018), an analysis of residential consumers metered data has been carried out to identify common consumption patterns leveraging an unsupervised analysis exploiting the extraction of duration curves to properly characterise the periodic consumption of each customer. In (Barak & Mokfi, 2019), a new framework tailored to the selection of the best clustering algorithm between a pool of algorithms to provide the best partitioning for a specific task has been presented, leveraging a multiple criteria decision-making strategy based on a multiple-indexes analysis and tested on 4 datasets with different characteristics. Finally, authors in (Ünlü & Xanthopoulos, 2019) propose a new strategy to automatically select the best partitioning for a given clustering task, optimising the weighted consensus between four different indexes calculated on a range of possible solutions.

Machine learning applications to cyclic manufacturing processes have been proposed in literature. (Kozjek et al., 2017) proposes a two phases data-analysis workflow to identify faulty conditions for the cyclic production process by analysing production signals along with their corresponding machine alarms, and the labels describing if a cycle brought to faulty condition or not. Thus, a decision tree is trained to extract the rules characterising failures. Then, a clustering analysis is used to identify the types of faulty conditions supporting operators in understanding the main causes of malfunctioning. However, they do not provide any general strategy on how to extract features from cycle based production processes, focusing on one specific case study. Instead, we are proposing a general feature engineering process to automatically characterise raw production signals independently from their nature and without any prior knowledge about the status of the machinery, i.e. without ground-truth labels. Then, differently from the state-of-the-art, we propose a new strategy to label each production cycles in an unsupervised fashion, mining latent patterns of the production process, and highlighting not only faulty conditions.

To sum up, the solution proposed in this paper enhances the state-of-the-

art methods by (i) introducing a production-cycle-aware feature engineering, aimed at (ii) discovering unsupervised production-cycle categories. To this aim, (iii) a self-tuning approach drives the automatic choice of optimal clustering techniques and parameters, whose results (i.e., the production-cycle categories) are (iv) described by a human-readable characterisation, hence providing (v) an assisted cluster-labelling process for domain experts without requiring machine-learning specific skills.

3. The UDaMP's architecture

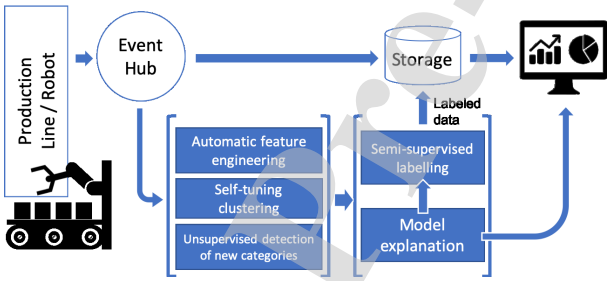


Figure 1: The UDaMP architecture with the main conceptual building blocks.

The main components of the UDaMP's engine and their functional connections are shown in Figure 1. UDaMP (Unsupervised Data-driven Methodology for Production-cycles Characterisation) provides semi-supervised data labelling to feed predictive analytics algorithms when ground-truth knowledge is not available. Currently, most of the predictive maintenance approaches rely on supervised algorithms, requiring ground-truth knowledge (i.e., class labels) of the phenomenon under analysis, however, such labels are often unavailable, at least in the short term. To address this issue, UDaMP includes a self-tuning semi-supervised data labelling pipeline, which automatically identifies new unseen class labels, and allows domain experts to drastically reduce their manual intervention by requiring the inspection of a very limited subset of representative samples for each new category. Each cluster is locally characterised through its own data statistics, to help domain experts in understanding its

content separately. The few representative label assignments are then exploited to automatically categorise the remaining samples. To this aim, an appropriate data partitioning is built on historical data by means of clustering algorithms, to automatically discover groups of data sharing common properties without requiring previous knowledge of their existence. Furthermore, UDaMP introduces a self-tuning strategy to automatically configure and select an optimal clustering algorithm, along with an automatic feature engineering technique.

To recap UDaMP consists of the following conceptual building blocks: (i) automatic feature engineering for manufacturing data events, (ii) unsupervised clustering for detecting possible new categories of data, (iii) self-tuning of clustering parameters, (iv) cluster characterization to provide explainability of the models, and (v) semi-supervised labelling.

To perform such data analytics tasks, the data collection block of UDaMP, named *event hub*, is designed to reliably route a virtually unlimited number of sensor measurements and log events from heterogeneous manufacturing-plant data sources at different rates. The *event hub* routes data to the UDaMP's analytics blocks. Both the results of the analytics and the long-term storage of the raw data are performed by a scalable high-performance NoSQL data store (implementation details are provided in Section 4). As an alternative, a scalable high-performance relational database can be used as well.

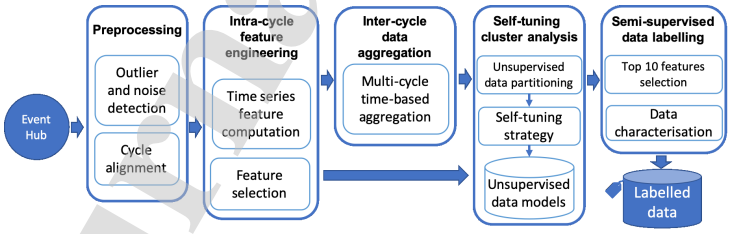


Figure 2: Detailed steps of the specific data analytic pipeline in UDaMP.

Figure 2 shows the detailed building blocks of the analytic pipeline: the *data preparation* steps (described in Section 3.1), the *self-tuning cluster analysis* (described in Section 3.2), and the *semi-supervised data labelling* (described in

Section 3.3).

3.1. Data preparation

This component performs three steps on data collected from the production plant. It includes *preprocessing*, *smart data computation*, and *data aggregation*, as detailed in the following. Algorithm 1 shows the data preparation process in details. The process takes in input the historical set of production cycles PC. Each $pc \in PC$ is a numerical vector, representing the signal collected in the manufacturing environment and describing one cycle of the production process. Also, the algorithm can be configured by changing parameters like s, that defines the number of splits in which the cycle has to be divided; w, the size of the aggregation window where patterns have to be analysed; th_f, that is the correlation threshold above which computed features would be discarded in the next phases of the analytic pipeline. Further details about these parameters will be provided in the next paragraphs. The output of the data preparation process is a set of features $f \in F$ characterising each production cycle over w.

Preprocessing. This step performs (i) outliers detection and removal, and (ii) missing value insertion. Specifically targeting cyclic manufacturing processes, UDaMP analyses the deciles of cycle lengths, and removes cycles belonging to the first and the last deciles, as they typically represent non-production or test cycles, as supported by domain-expert evidence (lines 1 to 4). Additionally, a cycle alignment task is performed when needed: value padding, i.e., repeating the value of the last cycle, is exploited until the cycle time slot is filled (line 5). This ensures a smoother analysis, thanks to a fixed-time structure, by means of the following *feature engineering* step.

Intra-cycle feature engineering. This step transforms raw time series, as collected from sensors in industrial plants, into time-independent feature sets. This strategy significantly reduces the dimensionality of the data, while preserving their informative content. the evolution in time of the considered time series. Each manufacturing cycle is divided into s splits over the time domain, with the goal of capturing intra-cycle data variability (line 6). An example of

Algorithm 1: Data prepatation (map-reduce syntax).

Input : Production Cycles PC; cycle_splits s; aggregation_window w; correlation threshold th_f;

Output: Cycles' features F;

`/* Preprocessing */`

1 $PC\_len \leftarrow PC$.map($pc \rightarrow$ length(pc));
2 $q10 \leftarrow$ computeQuantile(PC_len, 10%);
3 $q90 \leftarrow$ computeQuantile(PC_len, 90%);
4 $PCF \leftarrow PC$.filter($pc \rightarrow$ length(pc) $\geq q10$ **or** length(pc) $\leq q90$) ;
5 $PCP \leftarrow PCF$.map($c \rightarrow$ padCycle(c));

`/* Intra-cycle feature engineering */`

6 $PCS \leftarrow PCP$.map($pc \rightarrow$ splitCycle(pc, s));
7 $F \leftarrow PCS$.map($S \rightarrow$ computeStatisticalFeatures(S));
8 $F \leftarrow$ featureSelection(F, th_f) ;

`/* Inter-cycle data aggregation */`

9 **if** $w > 1$ **then**
10 $\quad F \leftarrow$ windowAggregation(F, w);
11 **end**
12 **return** F

split cycle is reported in Figure 3. Each split is then characterized by different statistical features (e.g. mean, standard deviation, quartiles, kurtosis, skewness, root mean squared error, sum of absolute values, number of elements over the mean, absolute energy, mean absolute change) to capture data variability within each split, with the split size being a customisable parameter (line 7). The feature computation produces a matrix F composed by m cycles and n features: each cycle is characterised by s splits, with each split being characterised by the different descriptive statistics, i.e., in our case a cycle is characterised by $s * 10$ features, since 10 different statistics are computed for each split. Having same-size splits is a choice of simplicity that was proven to work in our trial case. However, feature engineering can be successfully applied to splits of different sizes, since their purpose is to capture specific transient states and steady states

Algorithm 2: Window aggregation

Input : Per-cycle Features F; aggregation_window w;

Output: Aggregated features AF;

$AF \leftarrow \emptyset$;
$FW \leftarrow$ groupByWindow(F, w);
foreach fw ***in*** FW **do**
  $AFC \leftarrow \emptyset$;
  foreach fw_j ***in*** $\{fw_{i,j} | 1 \leq i \leq m\}$ **do**
    $min_{fw_j} \leftarrow \min fw_j$;
    $max_{fw_j} \leftarrow \max fw_j$;
    $mean_{fw_j} \leftarrow mean(fw_j)$;
    $std_{fw_j} \leftarrow std(fw_j)$;
    $\beta_{fw_j}, int_{fw_j} \leftarrow$ linearRegression($\{1, \ldots, m\}, fw_j$);
    $AFC \leftarrow AFC \cup \{min_{fw_j}, max_{fw_j}, mean_{fw_j}, std_{fw_j}, \beta_{fw_j}, int_{fw_j}\}$
  end
  $AF \leftarrow AF \cup AFC$;
end
return AF

in cyclic industrial processes, whose duration can vary over time. Finally, a feature selection phase is performed over (line 8). To this aim, UDaMP integrates the correlation-based approach, which evaluates the correlation of each couple of attributes, and removes those that are correlated the most, on average over all the (other) features, with the correlation threshold th_f being customisable.

Inter-cycle data aggregation. Most industrial processes are characterized by slowly-degrading effects, where the single cycle is extremely short with respect to the target degradation phenomena and its prediction horizon. Hence, predicting the condition of a specific cycle is not of interest to domain experts, whereas the focus is on longer periods, from hours to days, spanning over many production cycles. Thus, the aim of this step is to aggregate the intra-cycle features over longer inter-cycle time windows. The window size w is expressed as the number of cycles to be aggregated. If $w > 1$, then the cycle features are

aggregated (Algorithm 1 line 10). Moreover, Algorithm 2 shows in detail how the aggregation is performed. First, the features F are divided into sequential groups $fw \in FW$, each of size w, assuming that the input data are ordered by timestamp. Then, since intra-cycle features are related to each specific feature of each split of each cycle, the inter-cycle aggregation is computed separately for each feature. In particular, among the other statistics, a linear regression on the aggregation period fw is computed for each feature (Algorithm 2 line 10). Thus, the aggregation phase stores the slope and intercept coefficients of the regression, and the minimum, maximum, mean and standard deviation for each feature in each window of analysis fw (Algorithm 2 lines 5 to 12). The output is the aggregate matrix of features AF composed by $\hat{m} < m$ cycles, according to the size of the window, and $\hat{n}$ features, that correspond to the number of selected features in F multiplied by the number of statistics computed during the aggregation process. Please note that both the feature selection and the feature aggregation steps preserve human readability, hence keeping the approach transparent and its decisions easily accountable.

3.2. Self-tuning cluster analysis

The aim of this block is to automatically infer interesting, cohesive and well-separated groups of production cycles from the unlabelled data collected in the production plant. It integrates different *state-of-the-art clustering algorithms* and a *self-tuning strategy*. While the former block discovers a set of groups of production cycles characterised by similar properties given a specific setting of input parameters, the latter (i) automatically discovers the optimal input parameter setting for each algorithm and (ii) selects the algorithm by finding the optimal partition tailored to the data under analysis. Both contributions address popular challenges in applying machine learning techniques to real-world settings under the supervision of application-domain experts.

UDaMP currently integrates three state-of-the-art partitional clustering algorithms: (i) K-Means (Hartigan & Wong, 1979), (ii) Bisecting K-Means (Tan et al., 2005), and (iii) Gaussian Mixture (Lindsay, 1995).

The K-Means algorithm subdivides production cycles into k groups, where k is a user-specified parameter. Each group is represented by its centroid, computed as the average of all samples in the cluster. Although K-Means has a bias towards clusters with a spherical shape, it identifies a profitable data partition in a limited computational time in many real-life settings.

Bisecting K-Means applies the K-Means through a hierarchical and bisecting strategy. Instead of searching a global solution, it repeatedly focuses on a dataset portion to bisect it through the K-Means. The process is repeated until the (user-defined) k desired groups are met.

The Gaussian Mixture is a general iterative procedure to find groups of data originated by the same distribution. To find a good data partition, the algorithm estimates the mean and the standard deviation for each cluster and the sampling probability of each group, i.e., the probability that one of the N Gaussian distributions is used as a source of data. Similarly to the K-Means and Bisecting, the Gaussian Mixture algorithm requires the desired number of groups as an input parameter.

UDaMP includes a *self-tuning strategy* to automatically discover for each clustering algorithm an optimal input parameter setting It applies a well-known quality metric, named the Silhouette index (Rousseeuw, 1987), measuring how similar a production cycle is to its own cluster (cohesion) compared to other clusters (separation), by evaluating the appropriateness of the assignment of a production cycle to a cluster rather than to another one. Let $\mathbb{C} = \{C_1, \ldots, C_n\}$ be a set of n clusters, each one representing a group of production cycles. The Silhouette value for a given production cycle pc_i in a cluster $C_k \in \mathbb{C}$, given a distance measure d is computed as:

$$s(pc_i) = \frac{b(pc_i) - a(pc_i)}{\max\{a(pc_i), b(pc_i)\}}, \tag{1}$$

where $a(pc_i)$ is the average distance of production cycle pc_i from all other cycles in cluster C_k, i.e.,

$$a(pc_i) = \frac{1}{|C_k|} \Sigma_{pc_j \in C_k} d(pc_j, pc_i) \tag{2}$$

and $b(pc_i)$ is the lowest average distance from all other clusters, i.e.

$$b(pc_i) = \min_{C_l \in \mathbb{C}} \left(\frac{1}{|C_l|} \Sigma_{pc_j \in C_l} d(pc_j, pc_i)\right), \forall C_l \neq C_k. \quad (3)$$

The Silhouette ranges from -1 to +1, where a high value indicates that the production cycle is well matched to its own cluster and poorly matched to its neighbouring cluster. Negative and positive Silhouette values represent wrong and good placements, respectively. Hence, the ideal clustering algorithm splits the data into a set of clusters with production cycles characterised by a Silhouette value equal to 1. However, lower Silhouette values around 0.2-0.3 are already considered good values in real-life settings since real datasets are usually characterised by variable data distributions.

To automatically identify a good configuration of the input parameter for each clustering algorithm, UDaMP automatically analyses the trend of the harmonic average of the average Silhouette index (ASI) and the global Silhouette index (GSI) against the desired number of clusters K. ASI and GSI indexes (Cerquitelli et al., 2018) are defined as follow:

$$\text{ASI} = \frac{1}{N} \sum_{i}^{N} s(pc_i) \quad (4)$$

$$\text{GSI} = \frac{1}{|\mathbb{C}|} \sum_{C_k \in \mathbb{C}} \frac{1}{|C_k|} \sum_{pc_l \in C_k} s(pc_l) \quad (5)$$

For both indicators (ASI and GSI), higher values correspond to better partitioning. ASI is a measure of how appropriately data have been clustered overall, and is calculated by averaging the Silhouettes over the entire cluster set of records (N is the cardinality of the dataset), instead GSI takes into account the imbalance of the number of elements in each cluster, by penalising clusters with a large number of production cycles, which would otherwise have a predominant weight.

The harmonic average of ASI and GSI is exploited in UDaMP to take advantage of key aspects of both indicators.

The self-tuning step selects the value of K as the last value corresponding to an increasing trend of the harmonic average of ASI and GSI. Such process

is performed separately for each algorithm, hence obtaining different values of K and their corresponding Silhouette. The sorted Silhouette curves (i.e., the sorted Silhouette values computed for each production cycle) are then compared. The partition with the best overall Silhouette trend is selected as the clustering result.

To the best of our knowledge, the Silhouette index has never been exploited to automatically configure the input parameter of any clustering algorithms due to its high computational cost: all the pair distances among samples have to be computed. However, since a scalable approach to evaluate the Silhouette index, named Descriptor Silhouette (DS), has been recently proposed in (Ventura et al., 2019), it has been integrated into UDaMP to provide an innovative, efficient and effective self-tuning service.

3.3. Semi-supervised data labelling

To support domain experts in inspecting a very limited subset of representative samples, hence effectively reducing the efforts of such manual process while enhancing the understanding of the data, each cluster is locally characterised through:

(i) *The top 10 intra-cycle features*, to focus the domain-expert attention on the most relevant characteristics modelling each cluster of production cycles. To this aim, UDaMP uses a Classification and Regression Tree (CART) (Tan et al., 2005). The CART is built using the same inputs of the clustering algorithm. The cluster identifier assigned by the clustering process is selected as target label of the CART. The first 10 features used as splits in the CART tree nodes represent the most relevant features characterising the paths from the tree root to the leaves.

(ii) *Boxplot distribution of the top-10 features*, separately for each cluster, to characterise its content in terms of the most relevant properties. It represents a visual support to help application-domain experts to easily derive a specific meaning for each cluster. Few representative samples for each cluster are then manually inspected by domain experts to apply a label and verify the correctness

of the assignment. Their label assignments are then used for the remaining samples in each corresponding cluster.

4. The UDaMP technological solution

The design and development choices of UDaMP have been driven by an in-depth analysis of Industry 4.0 requirements carried out with different leading companies, from mechanical to robotics, from automation to food supplement production. Specifically, the following choices have been made to address the needs of modern manufacturing industries:

Reliability, availability, scalability, and manageability have been provided through the exploitation of state-of-the-practice technologies that reasonably pledge such properties. To provide the manageability, the UDaMP architecture has been deployed in a containerised environment, which significantly reduces the architectural management complexity. Furthermore, containerised blocks are typically horizontally scalable, as they can be easily instantiated dynamically by design, both on-premises and in cloud environments. In UDaMP each building block is encapsulated into a Docker container (Merkel, 2014), by providing a flexible loosely-coupled architecture.

To develop the event hub, Apache Kafka (Kreps et al., 2011) has been selected to guarantee a horizontally scalable, fault-tolerant, advanced message broker able to effectively deal with real-time applications requiring significant data throughput.

A specific challenge of Industry 4.0 data storage is to guarantee a good scalability in writing operations. To this aim, UDaMP exploits Apache Cassandra (Lakshman & Malik, 2010), a column-oriented NoSQL database, as data storage.

Finally, the self-tuning semi-supervised data labelling blocks are based on Apache Spark (Zaharia et al., 2012), along with its streaming extensions and the parallel machine learning library, MLlib (Meng et al., 2016). Spark Streaming is a state-of-the-practice, horizontally-scalable, high-throughput and fault-tolerant

framework for soft real-time Big Data analysis.

5. Experimental results

The experimental section is organised as follows. Section 5.1 describes the real-life Industry 4.0 case studies. Section 5.2 evaluates the benefits of the self-tuning clustering analysis block along with the semi-supervised data labelling task. Section 5.3 compares UDaMP to other state-of-the-art approaches.

All the experiments have been performed on an Intel Core i7 machine with 32 GB of RAM running Ubuntu 16.04 and a cluster of two nodes configured with Apache Kafka 1.0, Spark (and MLlib) 2.4.0, Docker 17.09, and Cassandra 3.11. The clustering algorithm implementations are those available in the Apache Spark MLlib (Meng et al., 2016) library, while to compute the Silhouette index UDaMP integrates Descriptor Silhouette (DS), as proposed in (Ventura et al., 2019).

5.1. Real-life case studies

UDaMP has been validated in the following digital factories to evaluate its effectiveness.

International white-goods company (CS1). UDaMP has been evaluated on data collected from sensors placed in a manufacturing plant of an international white-goods company. The production cycle involves using a nozzle to inject isolating foam. The nozzle has been sensorised to monitor the process. A predictive model of the overall system conditions is desired to predict possible alarms and future failures in the process. A variety of signals have been monitored including the temperature of chemicals involved, pressure of the liquids before injection, injection timing and quantity, ratio of the injected chemicals, etc.

Ground-truth labels characterising each production cycle or set of production cycles are not available, since their definition and assignment would be very time-consuming. To this aim, UDaMP has been used to support domain experts in quickly and easily defining class labels. The goal of CS1 is to predict

if a given set of monitored production cycles (e.g., a few hours of data) will trigger an unexpected event in a given time horizon (e.g., the next day). The foaming process is characterised by very slow degradation trends, hence, batches of approximately 170 production cycles of few minutes each are aggregated into one-day-long time windows, during the preprocessing step, as discussed in Section 3.1.

International robotics industry (CS2). UDaMP has been validated on a set of data describing motor cycles collected in a robotics industry, where a *RobotBox* has been executing a given cycle continuously. The goal is the detection and prediction of the engine transmission belt degradation over time. Electrical data of the engine were collected for 16,862 cycles to study the belt tensioning level. Using a slider installed on the robot, different levels of the belt tensioning have been observed. Assessing the level of the belt tensioning for each cycle is an extremely time-consuming task. Hence, UDaMP has been exploited to support the domain experts in quickly and easily deriving class labels for each cycle.

Figure 3 shows an example of the electricity consumption values for a given robot cycle lasting about 24 seconds with 11,967 sampled measurements (sampling period of 0.02 seconds). The engine axis is parallel to the ground, and the engine cycle has been set as follows. At the beginning the initial position of the motor is -500 degrees; it slowly reaches +90 degrees (20% of the maximum speed) and it maintains the position for 5 seconds; then it returns to -500 degrees at its maximum speed; finally it maintains the new position for 5 seconds. Figure 3 shows the current absorbed during an engine cycle in CS2. Highlighted splits within the cycle are those identified by UDaMP as the most relevant ones in the semi-supervised data labelling block. Segment 6 represents the slow descent phase of the engine payload. Segments from 12 to 16 show the current absorbed when the motor reaches and maintains the +90 degrees. Segment 21 shows the current absorbed at the end of the maximum speed phase to maintain the position of -500 degrees.

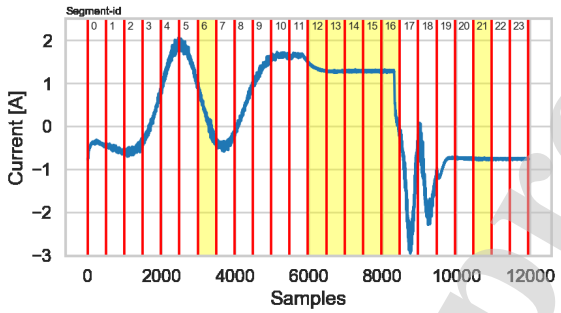


Figure 3: A production cycle sample for CS2. The most relevant splits identified by UDaMP are highlighted.

5.2. Evaluation of the self-tuning strategy and the semi-supervised data labelling

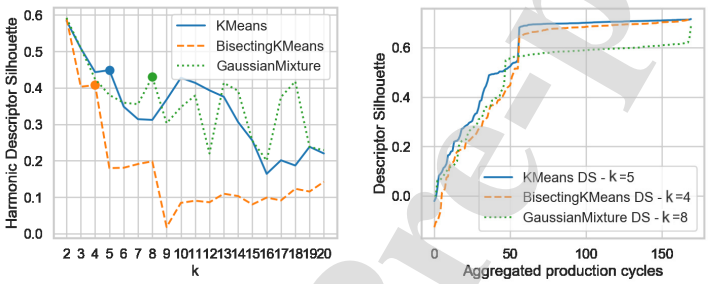
Intra-cycle feature engineering. For CS1, each cycle has been divided into 8 splits, whereas for CS2 the splits are 24, depending on the total length of the cycle. Then, intra-cycle features are computed as described in Section 3.1. The feature selection is applied with a 0.5 correlation threshold, which yields to 40 and 222 selected features for CS1 and CS2 respectively. Then, the aggregation step is executed for CS1, with 240 final features for each aggregated time window of 1 day. The data aggregation step is skipped for robot cycles in CS2 since each cycle needs to be characterised independently from the others.

Self-tuning cluster analysis. As discussed in Section 3.2, to automatically configure the cluster analysis, UDaMP examines the harmonic average of the Average Silhouette Index (ASI) and the Global Silhouette Index (GSI), called Harmonic Descriptor Silhouette (HDS), for k in the range $[2, 20]$.

Figure 4a and Figure 5a show the trend of HDS for each clustering algorithm in CS1 and CS2, respectively.

The selected configuration (i.e., the best value for the number of desired clusters) is highlighted with a coloured circle: CS1 yields to $k = 5$ for K-Means, $k = 4$ for Bisecting K-Means, and $k = 8$ for Gaussian Mixture; CS2 yields to $k = 6$ for K-Means, $k = 5$ for Bisecting K-Means, and $k = 7$ for Gaussian Mixture. To choose the best performing algorithm for the data under analysis

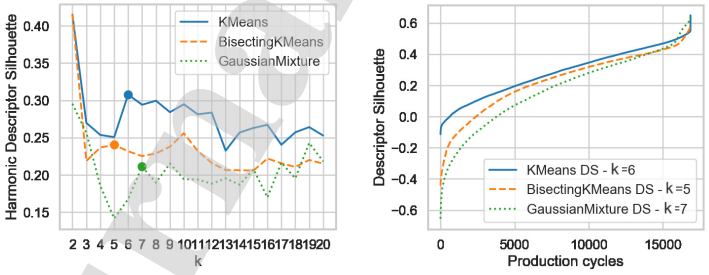
UDaMP compares the Descriptor Silhouette curves (Ventura et al., 2019) corresponding to the selected configuration for each algorithm, as shown in Figure 4b for CS1, and in Figure 5b for CS2. The algorithm with the highest values of Descriptor Silhouette is selected. For CS1 the best performing algorithm is K-Means with $k = 5$, whereas for CS2 the best one is K-Means with $k = 6$.



(a) Descriptor Silhouette curves for the selected k value for each algorithm.

(b) Descriptor Silhouette curves for the selected k value for each algorithm.

Figure 4: CS1: UDaMP self-tuning cluster analysis results



(a) Harmonic Descriptor Silhouette by varying k and clustering algorithms

(b) Descriptor Silhouette curves for the selected k value for each algorithm

Figure 5: CS2: UDaMP self-tuning cluster analysis results

Semi-supervised data labelling. UDaMP provides domain experts with

an interpretable representation of the automatically identified clusters of manufacturing cycles to help them in assigning a class label to each group. As described in section 3.3, UDaMP selects the top 10 features (by exploiting a CART) for better focusing on the most relevant characteristics under analysis. Figure 6 shows the boxplot of the selected features (listed in the x axis of each box-plot), separately for each cluster discovered in CS2. The selected features represent the most peculiar aspects of the manufacturing cycles.

From the box-plots in Figure 6 we observe that the first 5 clusters (from 0 to 4) are cohesive, well-separated, and well-balanced.

Cluster 5 instead, is characterised by a very sparse distribution of values, and it only includes 3 records. Indeed, we fixed the y axis range to keep all the other clusters comparable, and cluster 5 values are so different that they fall outside of the visible area.

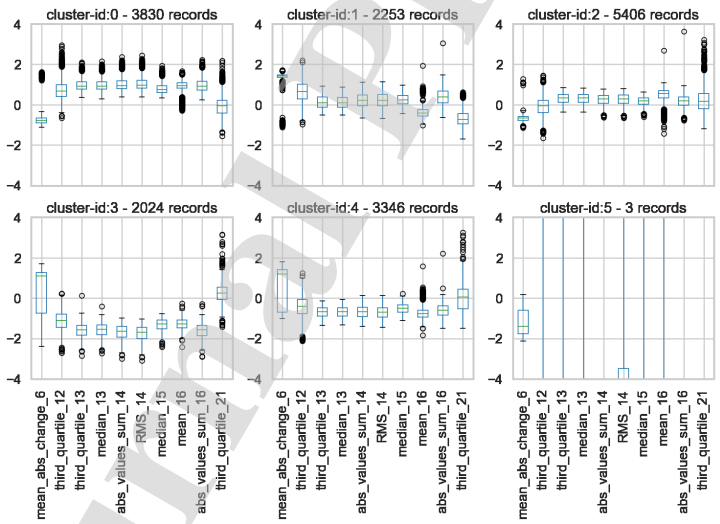


Figure 6: CS2: Data distribution of the top-10 relevant features separately for each group. The name of the features, reported on the x axis, include $< feature\_name > \_ < segment\_id >$

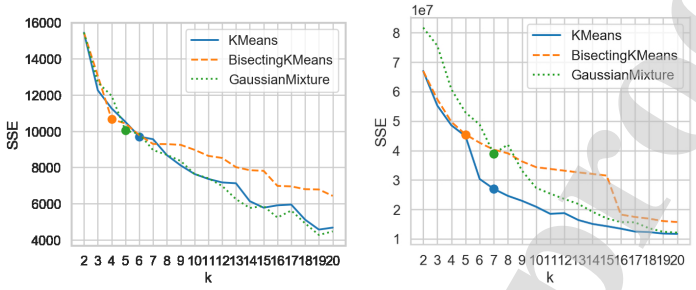
After the analysis of the aforementioned data distribution, domain experts from the involved companies confirmed that each group correctly models a spe-

cific belt tension level: (i) the selected segments (with ids 6, 12, 13, 14, 15, 16 and 21, also shown in Figure 3) correspond to the ones mainly affected by the belt tension, and (ii) no overlapping boxes appear in different groups. Thus, they are able to correctly understand the meaning of each group, i.e., link the cluster to a specific aspect of the physical phenomenon, and easily define the corresponding class label.

The main outcome of the semi-supervised knowledge-discovery process is the quick and informed class-label assignment for each cluster. It can be summarised as follows. The robot consumes a different amount of current (1) when it stops (split 12), (2) when it maintains the position of +90 degrees (splits 13-15), and (3) when it starts to move again at high speed (segment 16), based on the different levels of belt tension. Furthermore, when the engine is stopped in a position different than 0 degrees or starts to move, the effect of the level of belt tension over the absorbed current is higher w.r.t. when the engine is moving and the belt is held under tension. Then, in CS2, the clusters can be labelled by a domain expert as follows (from cluster 0 to 5, in order): (i) cluster 0 models the consumption of high positive current, (ii) cluster 1 models the consumption of medium positive current, (iii) cluster 2 models normal current consumption, (iv) cluster 3 models high negative and (v) cluster 4 models medium negative current consumption. Finally, (vi) cluster 5 collects outlier records.

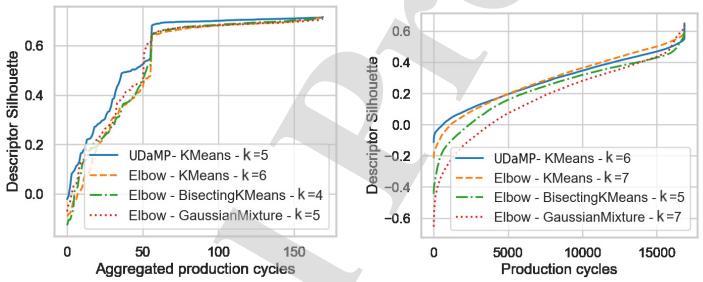
5.3. Comparison with the state-of-the-art approach

We consider the Elbow method (Satopaa et al., 2011), also known as knee approach, as the current state-of-the-art approach. It identifies the best value for the desired number of clusters by analysing the trend of a quality measure, such as the Sum of Squared Errors (SSE), against k. The value of k corresponding to the local minimum (knee) in the SSE trend is then selected, meaning that the gain from adding a new cluster is negligible, thus the reduction of the SSE is no longer valuable. Figure 7 shows the SSE curves against k for the three clustering algorithms in CS1 and CS2. The selected value of k for each algorithm is highlighted with a coloured circle.



(a) CS1 (b) CS2

Figure 7: Elbow method, SSE trends against k for all clustering algorithms, selected values of k are indicated by a filled circle.



(a) CS1 (b) CS2

Figure 8: Comparison of Descriptor Silhouette curves.

To compare the quality of the best clustering result found by UDaMP with the ones produced by the state-of-the-art method, the Descriptor Silhouette curves are exploited, as shown in Figure 8. Partitions selected by UDaMP are characterised by generally higher silhouette values, meaning a better inner-cohesiveness and a greater inter-separation, than the solutions found by the state-of-the-art method. In practice, UDaMP provides more reliable results, since its clusters better group together similar production cycles, and vice versa diverse production cycles are assigned to different clusters. We recall that the

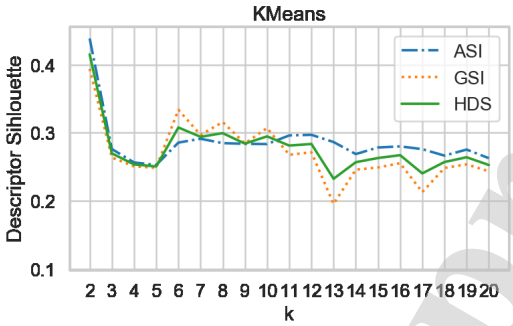


Figure 9: Descriptor Silhouette for ASI, GSI, and HDS against k: K-Means in CS2.

similarity is expressed on the basis of the features extracted in the data preparation step (Section 3.1).

In Figure 8b, we note that the Descriptor Silhouette curve for CS2 with $k = 6$ (value selected by UDaMP) is on average slightly lower than the one computed with $k = 7$ (value selected by the Elbow method). This is mainly due to UDaMP adopting the strategy of (i) taking into account the quality of the overall partitions in the clustering result (through ASI) while (ii) penalising large clusters (through GSI). To better clarify this issue, in Figure 9 we report the trends of the Average Silhouette Index (ASI), the Global Silhouette Index (GSI), and the harmonic average of ASI and GSI (HDS), against k, with the K-Means algorithm. Comparing the values for $k = 6$, selected analysing the HDS curve and $k = 7$, selected by the Elbow method, we notice that the ASI value for $k = 7$ is slightly higher, whereas the GSI is definitely much higher for $k = 6$. Thus, the clustering result with $k = 6$ provides more cohesive and well-separated groups then the result with $k = 7$. This highlights the possibility that a lower SSE might not represent better inner-cohesive and inter-separated groups, making the Elbow method a less reliable approach.

Moreover, the UDaMP architecture has been designed to be applied in industrial settings, where domain experts (who are not expected to be data scientists) require a clear vision of why a clustering result has been selected as the best

one. The Elbow method is a suitable solution for problems where the clustering reaches an evident knee in the SSE curve, which is not always the case: the SSE curve might slowly degrade without an evident knee. For example, the reasons why $k = 6$ has been chosen as the best clustering parameter for CS2 are definitely clearer by reading Figure 5a w.r.t. inspecting the SSE curve in Figure 7b.

6. Discussion

In this section, we analyse UDaMP's contributions and provide new horizons opened by the proposed work.

Unsupervised labelling. Currently, a major challenge in manufacturing intelligence is the absence of timely and precise class labels for training supervised predictive-maintenance classification techniques. The solution proposed by UDaMP provides assisted labelling of entire clusters of production cycles, hence it drastically reduces the manual intervention required by domain experts, since they inspect only a very limited subset of representative information. This is an intermediate step towards a totally unsupervised labelling process, which is part of the planned future works.

Applicability and impact. Many manufacturing processes entail a series of cyclic procedures. The proposed approach is specifically designed to exploit the cyclic characteristics during the feature engineering and aggregation phases. We presented experimental results from two meaningful real-world use cases. We deem them to be meaningful and general since they stem from a joint research project with leading international companies from different areas of manufacturing (electronic appliances and robotic automation). Having designed the proposed data pipeline based on such wide industrial settings makes us confident of its broader applicability to other cycle-based processes, beyond the two specific use cases, with a very limited effort. Furthermore, our framework attempts to put a step forward in the digitisation of the manufacturing industry. Among the expected impacts we consider lowering the costs of extract-

ing knowledge from the huge amount of data collected in modern production systems, and increasing the awareness of the domain experts in understanding the application of data analytics solutions to the manufacturing process.

7. Conclusions

This paper presents UDaMP, an integrated platform to provide self-tuning semi-supervised data labelling, enabling domain experts to capitalise on predictive maintenance analytics by simplifying the cumbersome process of manual data labelling. The proposed solution has been tested on two different real-life use cases, showing its effectiveness in automatically inferring knowledge from data.

Future directions of this research work include: (i) the integration of anomaly-detection techniques exploiting scalable one-class classifiers; (ii) combining concept-drift detection techniques with one-class classifiers to automatically discover new types of production cycles, thus enriching the expertise of domain experts through knowledge inference from collected data.

8. Acknowledgements

The research activities leading to the results presented in this paper have received funding from the following entities.

- European Commission under the H2020-IND-CE-2016-17 program, FOF-09-2017, Grant agreement no. 767561, "SERENA" project, VerSatilE plug-and-play platform enabling REmote predictive mainteNAnce.
- `SmartData@PoliTO` center, Politecnico di Torino, Italy.

References

Apiletti, D., Barberis, C., Cerquitelli, T., Macii, A., Macii, E., Poncino, M., & Ventura, F. (2018). istep, an integrated self-tuning engine for predictive maintenance in industry 4.0. In *IEEE*

ISPA/IUCC/BDCloud/SocialCom/SustainCom 2018, Melbourne, Australia, December 11-13, 2018 (pp. 924–931).

Barak, S., & Mokfi, T. (2019). Evaluation and selection of clustering methods using a hybrid group mcdm. *Expert Systems with Applications*, *138*, 112817. URL: http://www.sciencedirect.com/science/article/pii/S0957417419305135. doi:https://doi.org/10.1016/j.eswa.2019.07.034.

Cerquitelli, T., Chicco, G., Corso, E. D., Ventura, F., Montesano, G., Armiento, M., González, A. M., & Santiago, A. V. (2018). Clustering-based assessment of residential consumers from hourly-metered data. In *2018 International Conference on Smart Energy Systems and Technologies (SEST)* (pp. 1–6). doi:10.1109/SEST.2018.8495863.

Chen, L.-F., & Chien, C.-F. (2011). Manufacturing intelligence for class prediction and rule generation to support human capital decisions for high-tech industries. *Flexible Services and Manufacturing Journal*, *23*, 263–289.

Chien, C.-F., Chen, Y.-J., & Peng, J.-T. (2010). Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle. *International Journal of Production Economics*, *128*, 496–509.

Chien, C.-F., Hsu, C.-Y., & Chen, P.-N. (2013). Semiconductor fault detection and classification for yield enhancement and manufacturing intelligence. *Flexible Services and Manufacturing Journal*, *25*, 367–388.

Davis, J., Edgar, T., Porter, J., Bernaden, J., & Sarli, M. (2012). Smart manufacturing, manufacturing intelligence and demand-dynamic performance. *Computers & Chemical Engineering*, *47*, 145 – 156. URL: http://www.sciencedirect.com/science/article/pii/S0098135412002219. doi:https://doi.org/10.1016/j.compchemeng.2012.06.037. FOCAPO 2012.

D'silva, G. M., Khan, A., Gaurav, & Bari, S. (2017). Real-time processing of iot events with historic data using apache kafka and apache spark with

dashing framework. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*. doi:10.1109/RTEICT.2017.8256910.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*, 100–108.

Kozjek, D., VrabiÄ, R., Kralj, D., & Butala, P. (2017). Interpretative identification of the faulty conditions in a cyclic manufacturing process. *Journal of Manufacturing Systems*, *43*, 214 – 224. URL: http://www.sciencedirect.com/science/article/pii/S0278612517300304. doi:https://doi.org/10.1016/j.jmsy.2017.03.001. High Performance Computing and Data Analytics for Cyber Manufacturing.

Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: a distributed messaging system for log processing.

Lakshman, A., & Malik, P. (2010). Cassandra: A decentralized structured storage system. *SIGOPS Oper. Syst. Rev.*, *44*, 35–40. URL: http://doi.acm.org/10.1145/1773912.1773922. doi:10.1145/1773912.1773922.

Lee, J., Kao, H.-A., & Yang, S. (2014). Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia CIRP*, *16*, 3 – 8. Product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems.

Lindsay, B. G. (1995). Mixture models: Theory, geometry and applications. *NSF-CBMS Regional Conference Series in Probability and Statistics*, *5*, i–163. URL: http://www.jstor.org/stable/4153184.

Marques, G., Pitarma, R., M. Garcia, N., & Pombo, N. (2019). Internet of things architectures, technologies, applications, challenges, and future directions for enhanced living environments and healthcare systems: A review. *Electronics*,

8, 1081. URL: http://dx.doi.org/10.3390/electronics8101081. doi:10.3390/electronics8101081.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., Xin, D., Xin, R., Franklin, M. J., Zadeh, R., Zaharia, M., & Talwalkar, A. (2016). Mllib: Machine learning in apache spark. *J. Mach. Learn. Res.*, *17*.

Merkel, D. (2014). Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, *2014*. URL: http://dl.acm.org/citation.cfm?id=2600239.2600241.

Park, J., Park, C., Kim, J., Cho, M., & Park, S. (2019). Adc: Advanced document clustering using contextualized representations. *Expert Systems with Applications*, *137*, 157 – 166. URL: http://www.sciencedirect.com/science/article/pii/S0957417419304762. doi:https://doi.org/10.1016/j.eswa.2019.06.068.

Ribeiro, M., Grolinger, K., & Capretz, M. A. M. (2015). Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. doi:10.1109/ICMLA.2015.152.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53 – 65.

Satopaa, V., Albrecht, J., Irwin, D., & Raghavan, B. (2011). Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops* (pp. 166–171). doi:10.1109/ICDCSW.2011.20.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Ünlü, R., & Xanthopoulos, P. (2019). Estimating the number of clusters in a dataset via consensus clustering. *Expert Systems with Applications*, *125*, 33 – 39. URL: http://www.sciencedirect.com/science/article/pii/S0957417419300892. doi:https://doi.org/10.1016/j.eswa.2019.01.074.

Ventura, F., Proto, S., Apiletti, D., Cerquitelli, T., Panicucci, S., Baralis, E., Macii, E., & Macii, A. (2019). A new unsupervised predictive-model self-assessment approach that scales. In *2019 IEEE International Congress on Big Data (BigData Congress)* (pp. 144–148). IEEE. doi:10.1109/BigDataCongress.2019.00033.

Wang, S., Wan, J., Zhang, D., Li, D., & Zhang, C. (2016). Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, *101*. doi:https://doi.org/10.1016/j.comnet.2015.12.017. Industrial Technologies and Applications for the Internet of Things.

Yan, J., Meng, Y., Lu, L., & Li, L. (2017). Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance. *IEEE Access*, *5*, 23484–23491. doi:10.1109/ACCESS.2017.2765544.

Yao, Y., Xiao, Z., Wang, B., Viswanath, B., Zheng, H., & Zhao, B. Y. (2017). Complexity vs. performance: Empirical analysis of machine learning as a service. In *Proceedings of the 2017 Internet Measurement Conference* IMC '17 (pp. 384–397). New York, NY, USA: ACM. doi:10.1145/3131365.3131372.

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S., & Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *NSDI'12*.

ORCID Information

Tania Cerquitelli: 0000-0002-9039-6226
Francesco Ventura: 0000-0003-3398-8265
Daniele Apiletti: 0000-0003-0538-9775
Elena Baralis: 0000-0001-9231-467X
Enrico Macii: 0000-0001-9046-5618
Massimo Poncino: 0000-0002-1369-9688

- Semi-supervised labelling of production cycles for predictive maintenance.
- Wide applicability to any cycle-based production process without ground truth labels.
- Exploitable by domain experts thanks to transparent self-tuning techniques.

CRediT author statement

Tania Cerquitelli: Conceptualization, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration; **Francesco Ventura:** Methodology, Investigation, Software, Writing - Original Draft, Writing - Review & Editing; **Daniele Apiletti:** Writing - Original Draft, Writing - Review & Editing, Supervision; **Elena Baralis:** Project administration, Funding acquisition, Supervision; **Enrico Macii:** Project administration, Funding acquisition; **Massimo Poncino:** Project administration, Funding acquisition

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: