Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 54-1 (2021) 1248–1253

Data-Driven State Detection for an asset working at heterogenous regimens ⋆

Domenico Daniele Nucera * **Walter Quadrini** *
Luca Fumagalli * **Marcello Paolo Scipioni** **

* *Department of Management, Economics and Industrial Engineering, Politecnico di Milano, p.zza Leonardo da Vinci 32, 20133 Milan, Italy (e-mails: domenico.nucera@polimi.it; walter.quadrini@polimi.it; luca1.fumagalli@polimi.it).*
** *Fincons SPA, via Torri Bianche 10, 20871 Vimercate (MB), Italy (e-mail: marcello.scipioni@finconsgroup.com).*

Abstract: The current trend of industrial digitalization paved the way to Machine Learning applications which are adding value to data coming from the assets. In this context, the case study of a State Detection in an asset characterized by heterogeneous working regimens is proposed, with the aim of automatically recognizing the type of the ongoing production and of identifying its different operating conditions. The activity is executed by exploiting the data available on the asset controller and applying and comparing two different clustering algorithms, namely K-Means and HDBSCAN. The paper describes hence the application case and the adopted approaches, while providing insights on the most preferable choice for any of the two objectives, in order to pave the ground for condition-based maintenance activities.

Keywords: Production activity control, Quality assurance and maintenance, State Detection, Clustering, K-Means, HDBSCAN

1. INTRODUCTION

In the last decades, the manufacturing sector has undergone an impressive digital transformation: the decrease of costs of microcontrollers and sensors, together with the evolution of wireless and wired communication devices and protocols, paved the ground to a blossoming of data sources within the single production machines. The dissemination of success stories related to the data analytics (which led to the famous sentence "data are the new oil", as debated by (Humby, 2006)) and the 2010 estimation that manufacturing sector generated about two hexabytes of data per year (Manyika et al., 2011), led industrial associations and consortia to agree on guidelines and de-facto standards to help manufacturing companies exploiting the value added by data, like IIRA (Lin et al., 2015) and RAMI4.0 (Zezulka et al., 2018), pursuing pre-existent standards devoted to the process management (e.g. Purdue Enterprise Reference Architecture (Williams, 1994)) and data layering (e.g. IEC 62264 (Commission et al., 2016)). A recent work by (Zhong et al., 2017) witnessed the spread of this trend with respect to the different technologies and areas of application in a manufacturing environment: the picture that emerges from this analysis shows a pervasive diffusion of the data-driven technologies in all the manufacturing processes, from the field level to the decision-making one. This paper work sits in the asset management field, defined by (Davis, 2016) as "the coordinated activity of an organisation to realize value from assets", which, on the other hand, have been defined by (Mattioli et al., 2020) as "entities having potential or actual value for an organisation". The authors of the aforementioned work have also divided the asset management lifecycle in five steps: (1) Acquisition, (2) Deployment, (3) Utilisation and Monitoring, (4) Maintenance, Repair and Overhaul and (5) Retirement. This work takes into consideration the activities implied by the third step in order to improve the effectiveness of the fourth one: as efficiently depicted by (Kammoun and Rezg, 2018), a clustering analysis based on the operational data can result in dramatical benefits for the further analysis targeting maintenance purposes. Machine Information Management Open System Alliances (MIMOSA) designed the OSA-CBM, stating a standard practice to frame the data flow in Condition Based Maintenance applications, comprising seven modules sets (Raheja et al., 2006), among which *condition-monitoring modules* are required to compare the analysed data with expected values and ranges, thus performing a State Detection.

This work covers an application case where this task is not trivial due to the lack of domain-knowledge on the physical asset under analysis and due to its operational characteristics, involving heterogeneous working regimens. Furthermore, it presents how Machine Learning can provide new powerful tools in the form of clustering algorithms, employing the acquired data for a Data-Driven State Detection. Data generated from the machine will be hence used to automatically recognize the type of production being executed.

This paper is structured as follows: in Section 2 previous works related to the field of Machine Learning and Industrial Engineering will be covered; Section 3 will present

⋆ This work is supported by Lombardy funded project SMART4CPPS (ID: 236789 CUP: E19I18000000009)

10.1016/j.ifacol.2021.08.149

the scenario characterizing the case study, while Section 4 will delve into the details of the clustering algorithms used; lastly, Section 5 will cover the results of the analyses carried out on the available data.

2. BACKGROUND

A well-known overview about deriving maintenance information from machine data was provided by (Jardine et al., 2006), listing the main techniques to support maintenance decisions distinguishing among statistical-based, AI-based and model-based approaches. In 2018, (Cattaneo et al., 2018) published a related work, somehow updating the aforementioned one with a structured overview focusing on the intersections between the topics of industrial engineering and Machine Learning (ML): basing on the distinction made by (Hastie et al., 2001), the ML techniques are divided in *Supervised Learning Methods*, whose goal is to predict an output over an input, and in *Unsupervised Learning Methods*, whose goal is to infer a structure, a path, or a series of sets, in a given series of observations. Among the *Supervised Learning Methods*, (Cattaneo et al., 2018) cite the following ones as most widely used in the manufacturing domain:

- *Linear Methods for Regression*, where a regression function is derived from the training dataset and used to predict an outcome from a subsequent input.
- *Linear Methods for Classification*, where the training dataset is a-priori divided into labelled regions and the subsequent inputs give, as an outcome, the pre-identified region they are supposed to belong to.
- *Decision Tree Based Methods*, where the attributes of an input are compared, following a tree-shaped graph in order to select the corresponding leaf node which will constitute the output.
- *Neural Networks*, where, according to (Hastie et al., 2001), through trial-and-error iterations, correlation between input and output pairs are constituded to build the network which will be used to predict unknown output from a new dataset.

For what concerns the *Unsupervised Learning Methods*, Cattaneo et al. (2018) list the following ones:

- *Principal Component Analysis*, where possibly correlated attributes are combined to retrieve a lower-dimension set of uncorrelated ones, able to still describe the previous attributes.
- *Association Rules*, where the joint values of different variables which appear most frequently are considered in order to predict combinations of attributes.
- *Cluster Analysis*, where the method aims at grouping the input elements into subsets or "clusters", such that the elements in each cluster are closer to the other elements of the same subset with respect to the ones belonging to other clusters.
- *Self-Organising Maps*, defined by (Hastie et al., 2001) as a clustering where the original high-dimension observations are mapped-down onto a two-dimensional coordinate system.

Even if non-exhaustive, the listed methods provide a good representation of the scenario engineers face when implementing ML-based methods to the data gathered from the machines, orienting their design considering the methods' requirements and purposes. A notable mention about the *Supervised Learning Methods* requirements relies on the availability of a training dataset where the training outputs are stored an labelled, with a precise and systematic relation with the correspondent inputs. It should also be noted that the depicted methods state the requirements and purposes, but several different implementations exist and are reported in scientific literature. In the presence of data gathered directly from measurements on industrial asset, target labels on which *Supervised Learning Methods* can be trained are rarely found. In these cases, a data mining activity is required to extrapolate knowledge from the signals available. In (Kusiak and Verma, 2011) this activity is executed through domain knowledge, while (Strachan et al., 2006) used the K-Means clustering algorithm to identify the conditions of Circuit Breakers. In fact, in this context, clustering analysis provides a technique to identify different states in an asset lifecycle. Several algorithms have been proposed in the literature and recent reviews provided by (Saxena et al., 2017) and (Mittal et al., 2019) categorize clustering techniques according to their inner mechanics, making a distinction between hierarchical, partition-based and density-based approaches.
K-Means is a partition method with a proven track of successful applications. The limitation of having to provide the number of partitions as an input can be regarded as a way to obtain better results when a certain level of starting knowledge is provided. Among the density based approaches, DBSCAN is regarded as able to recognise arbitrary shaped clusters, making it a powerful technique in many applications. However, according to (Campello et al., 2013), it assumes clusters to have uniform densities. This hypothesis is not always guaranteed in real application cases, in which certain conditions, sometimes the ones of interest, arise less frequently. The question of which clustering technique can be more appropriate in an industrial application remains open, but different algorithms can be considered as valid options according to the objective of our case study.

3. SCENARIO

The application case for this work has been conducted in a company world leader in the market of luxury fabrics printing. Among the several departments and assets contributing to the production, a textile steamer machine has been identified for the study case, given the criticality of the dye-fixing processing in the internal production chain and the availability of an OPC-UA (Mahnke et al., 2009) server on board of the machine that allows the deployment of a data mining platform, which has been deployed according to Figure 1, where a legacy OPC-UA routing software (namely Kepware, already analysed by (Border, 2018)) has been interfaced with a Elasticsearch-Kibana stack (Bajer, 2017) through a Kakfa middleware (Le Noac'h et al., 2017) in order to grant robustness and decouple data analysis from data generation (Quadrini et al., 2021). All the interfaces among these modules have been managed through the StreamSets environment.

The obtained data only contain the signals internally produced by the machine itself (e.g. temperatures, steam fluxes, motors speeds, energetic consumption) and do not

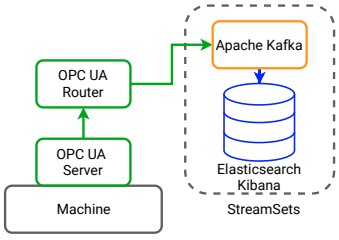

Fig. 1. Deployed architecture

allow by themselves to understand the precise machine status with respect to the ongoing production recipe. Furthermore:

(1) The raw materials arrive to the machine with an attached file declaring the nominal recipe the material has to undergo; then, the operator manually enters the working parameters into the Human-Machine Interface (HMI) of the machine, deviating some of them from the nominal recipe whenever required by the intrinsic characteristics of the fabric.
(2) The Enterprise Resource Planning (ERP) software is updated through a parallel informative system, in an asynchronous way.

The first consideration implies that every anomaly detection system used to trigger a CBM method would result in false positives whenever an operator sets machine parameters resulting in a recipe different from the nominal one. The second consideration leads to the impossibility to apply *Supervised Learning Methods*, given the asynchrony between the timestamp derived from the machine data and the eventual groundtruth given by the ERP system. These considerations led the authors to opt for clustering algorithms to identify the actual machine operating conditions in order to achieve the third step from the OSA-CBM standard. With the available signals, the authors also intend to use a clustering algorithm for the recognition of the type of production being executed by the operators, without having them manually entering it in the HMI.

4. ADOPTED CLUSTERING ALGORITHMS

In this work K-Means and HDBSCAN are going to be employed and compared. The implementation of K-Means and other techniques like PCA and Silhouette Coefficient evaluations are provided by (Pedregosa et al., 2011), while the HDBSCAN implementation results from (McInnes et al., 2017). In the following subsections both algorithms will be described, considering a set of n data points $\mathcal{X} = \{x_1, ..., x_n\}$ and a distance measure d.

4.1 K-Means

K-Means is an algorithm which aims to partition data points of a dataset into n_{cl} sets, according to (MacQueen et al., 1967). The algorithm thus requires to be provided in input the parameter n_{cl}, representing the number of clusters in which we intend to partition the data. It works by defining n_{cl} centroids $c_1, ..., c_{n_{cl}}$, representing their own clusters $\mathcal{C}_1, ..., \mathcal{C}_{n_{cl}}$. Its objective is to minimize the following objective function $\mathcal{F}$:

$$\mathcal{F} = \sum_{j=1}^{n_{cl}} \sum_{x_i \in \mathcal{C}_j} d(x_i, c_j)$$

representing the total sum of distances between data points with their respective centroids.
It starts by generating n_{cl} *centroids* $c_1, ..., c_{n_{cl}}$ with random coordinates and assigns each data point to its nearest centroid minimising its euclidean distance d. The procedure is hence iterated (recomputing every centroid coordinates upon the average coordinates of its near point) and stopped after the convergence is reached.

4.2 HDBSCAN

HDBSCAN was introduced by (Campello et al., 2013) as an extension of the DBSCAN algorithm, in which a hierarchy of DBSCAN* clusterings is built for varying ε, where DBSCAN* is an adaptation of standard DBSCAN which removes the notion of border points (McInnes and Healy, 2017). In this work we are going to employ a variant named HDBSCAN($\hat{\varepsilon}$) introduced by (Malzer and Baum, 2020). As we will see, it relies on an additional parameter to avoid the extraction of an excessive amount of clusters in the last flattening phase. Over the course of the work we will still refer to the algorithm employed as HDBSCAN. HDBSCAN relies on an alternative distance measure called *mutual reachability distance* d_{mreach} which, given a distance measure d and the parameter m_{pts}, defined the distance between two points x_p and x_q as:

$$d_{mreach}(x_p, x_q) = max\{d_{core}(x_p), d_{core}(x_q), d(x_p, x_q)\}$$

where $d_{core}(x_i)$ of a point x_i is the distance from its m_{pts}-nearest neighbour, with x_i included. In this, when considering a point in a low-density region, its *mutual reachability distance* from a possible core point will be bounded by its core distance $d_{core}(x_i)$, favouring it to be considered as noise due to its isolation. Once this measure is defined, it is possible to define a Mutual Reachability Graph having data points from $\mathcal{X}$ as vertices, connected by edges weighted according to to their *mutual reachability distance*. A Minimum Spanning Tree can be constructed from the Mutual Reachability Graph, and the dendogram resulting from sorting the mutual reachability distances can be used to discover clusters. Given the dendogram it is in fact possible to retrieve the clusters obtained using DBSCAN* by imposing a ε as a horizontal cut value, leaving as noise all the clusters consisting of less than m_{pts} (Malzer and Baum, 2020). With the aim of identifying clusters with different levels of density, the HDBSCAN flattening procedure results in a more refined approach. It proceeds by removing edges in decreasing order of *mutual reachability distance*, thus obtaining a split at every removal. If more edges have the same weight, we remove them at the same time. After a split, a connected subcomponent is evaluated by considering the number of vertices inside it:

- If the number of its vertices is fewer than a smoothing parameter m_{clSize}, then the connected subcomponent is deemed *spurious*, and we don't consider it as a cluster.

- If instead the number of vertices in a connected subcomponent is $\geq m_{clSize}$, we still consider it as a cluster.

After a removal, it is thus possible for a previously existing cluster:

- to just shrink, having lost some vertices now belonging to *spurious* subcomponents of the graph.
- to disappear for having split in several *spurious* subcomponents.
- result in possibly more than one cluster, in case after the split at least two of the new connected subcomponents are not considered *spurious*.

The result is thus a cluster tree in which at certain values of ε a cluster is born, and then at a lesser value of ε its life can be considered ended, either because it becomes *spurious* or because it splits into new clusters, considered its children. Given a cluster $\mathcal{C}_i$, $\varepsilon_{max}(\mathcal{C}_i)$ as the value ε at which $\mathcal{C}_i$ appears, and $\varepsilon_{min}(\mathcal{C}_i)$ as the value at which its lifetime ends are defined. From now on this formulation diverges from the one proposed by (Campello et al., 2013) and follows (Malzer and Baum, 2020), introducing an additional term $\hat{\varepsilon}$ accounting for a smoothing effect in the optimal cluster selection along the cluster tree. In fact, at this point different clustering choices in the cluster tree are available. For example, one could choose to select all the leaves in the tree, resulting in a clustering potentially characterized by a considerable quantity of different labels. Selecting instead a parent in the tree would result in its children being merged in a unique cluster, thus characterized by a unique label. To execute the clustering choice, the concept of *Epsilon Stability* of a cluster $\mathcal{C}_i$ is defined as:

$$ES(\mathcal{C}_i) = \begin{cases} \dfrac{1}{\varepsilon_{max}(\mathcal{C}_i)}, & \text{if } \varepsilon_{max}(\mathcal{C}_i) > \hat{\varepsilon} \\ 0, & \text{otherwise} \end{cases}$$

An algorithm is thus run leading to a clustering choice maximizing the overall *Epsilon Stability* from the cluster tree. The procedure can be found in detail in (Malzer and Baum, 2020).

5. APPLICATION RESULTS

For the purposes of this work, we intend to cluster the data points coming from the machine samples into groups according to the machine state and the type of production being processed. Among the signals coming from the machine's OPC-UA server, 6 features have been identified, anonymized as *feature1*, *feature2*, *feature3*, *feature4*, *feature5*, *feature6*. All of them have been standardized to have zero mean and standard deviation equal to one. The following information are relevant to the application case:

- The first two features do not reflect values of sensors installed on the machine. They instead account for machine settings entered by the operators to control its behaviour. Because of this, they happen to be discrete signals, having each one a limited amount of values.
- The third and fourth features are sampled from the machine, and are supposed to reflect the type of production being executed, meaning that their value is considered related with the type of operation carried on by the operator.
- The last two features, the fifth and the sixth ones, come from sensors installed on the machine, but their value is not directly influenced by the production type. Instead, they measure variables which can provide useful information regarding the machine behaviour.

The analyses can be thus divided into two different objectives with two different subsets of the original features:

- A clustering aimed at recognizing the production type according to a dataset composed by the first four features, which are directly supposed to account for the type of production being executed.
- A State Detection activity, always executed resorting to clustering techniques, aimed at identifying the different states the machine goes through. This clustering activity is thus based on signals directly coming from sensors installed on the machine, and thus on a dataset with only the last four features.

5.1 Production type recognition

To recognize the type of production we run the K-Means algorithm. To select the number of clusters to be provided in input, the foreground sits in the information that during the recording of the activity 5 different production types occurred. Since it's supposed that the machine can undergo additional phases unrelated to a specific type of production process, we ran the K-Means algorithm with the n_{cl} parameter ranging from four to nine. The silhouette coefficient, formulated by (Rousseeuw, 1987), has been hence evaluated for the different clusters. Table 1 depicts the results, highlighting that the coefficient does not increase relevantly beyond $n_{cl} > 7$. Aiming at clarifying the model, the clustering option with seven labels is selected, among which it becomes possible to recognise the five recorded production types. The remaining two labels account for a period in which the asset was set to operate in an anomalous regimen and an experimental activity which occurred during the sampling interval.

Table 1. K-Means silhouette coefficients for production recognition

n_{cl}	4	5	6	7	8	9
sc	0.868	0.869	0.897	0.945	0.944	0.948

5.2 State Detection

Regarding the State Detection, no a-priori knowledge on the conditions in which the machine can operate was available. HDBSCAN has been hence run on the dataset composed by *feature3*, *feature4*, *feature5* and *feature6* with the following parameters: m_{pts} set to 80, m_{clSize} set to 5 and $\hat{\varepsilon}$ set to 0.4. The resulting run of the algorithm outputs nine different clusters, with a silhouette coefficient of 0.646. To visualize the results of the clustering activity, a PCA on the dataset has been executed and the first three components, which cumulatively explain 90% of the total variance in the dataset, have been considered as well representative of the overall information content in the

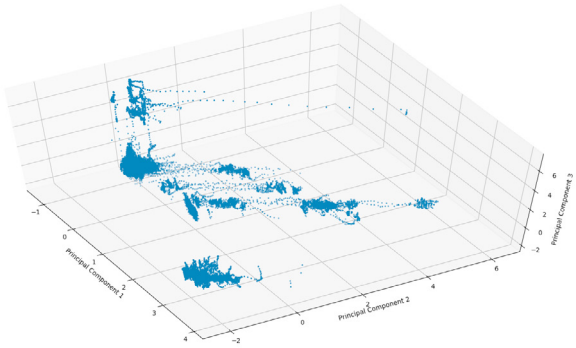

Fig. 2. The first three principal components from the State Detection dataset

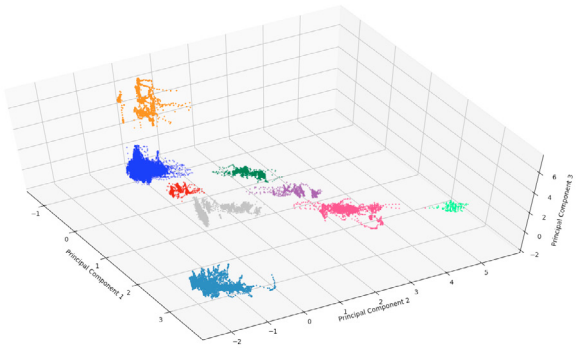

Fig. 3. HDBSCAN labelling on the State Detection dataset

dataset. A 3D scatterplot of the dataset in this new coordinate system is shown in Figure 2. The same scatterplot with the clustering results from the HDBSCAN execution can be found in Figure 3. As it can be seen, the main regions in which the samples are condensed are correctly isolated. Points missing from the previous plot have been identified as noise by the algorithm. Among the identified clusters an analysis with Domain Knowledge experts from the company was executed. As in Figure 3, the uppermost cluster, the one characterized by the highest level of the third principal component was recognized as a faulty behaviour. Clusters characterized by an increasing level of the second principal component regard the introduction of specific tissues inside the asset, whose characteristics impose a different operating condition due to the tissues' chemical components. To provide a comparison, K-Means has been executed with parameter n_{cl} ranging from four to ten. Silhouette coefficients from the runs can be seen in Table 2: the resulting coefficients suggest a clustering with six labels but, as Figure 4 shows, its partitioning is hardly comparable with the one provided by HDBSCAN. In fact the orange cluster accounts for two regions which can be visually considered as separated, while the operating region in the middle of the figure is split in halves and merged with the green clusters. At the same time the operating region on the right of the figure is still considered as belonging to the light green cluster. The K-Means clustering with n_{cl} equal to nine, the same number of clusters identified by HDBSCAN, can be found in Figure 5. In this case the underperforming result suggested by the silhou-

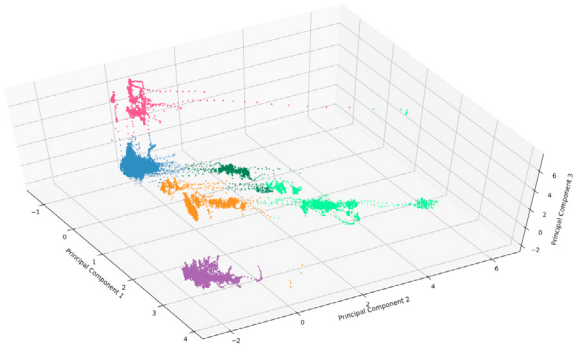

Fig. 4. K-Means clustering ($n_{cl} = 6$) on the State Detection dataset

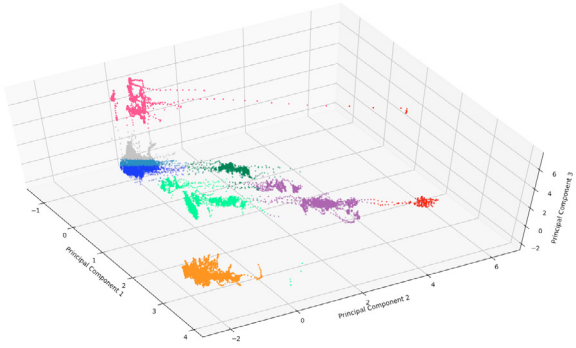

Fig. 5. K-Means clustering ($n_{cl} = 9$) on the State Detection dataset

ette coefficient is confirmed by visual inspection, in which an operating region clearly visually identifiable is split into three different clusters, while others are merged. It is possible to conclude that HDBSCAN allows to correctly isolate and identify different data points coming from sensors installed on the machines. In fact, a context with clusters posed at different distances, with varying sizes and non-trivial shapes can lead us to think a traditional distance method like K-Means as unable to execute an efficient State Detection, even when contrarily suggested by indicators like the silhouette coefficient.

Table 2. K-Means silhouette coefficients for State Detection

n_{cl}	4	5	6	7	8	9	10
sc	0.726	0.744	0.753	0.496	0.501	0.481	0.463

6. CONCLUSION AND FUTURE WORKS

This work addressed the case of an industrial asset working at different regimens, which resulted in two main peculiarities. On the one side, the possibility of learning from the data the different types of production being executed is an opportunity which can relieve the operators from the activity of manually entering into a register the type of operation they are working on. On the other side, the desire of designing a condition based maintenance solution for the asset requires a State Detection to identify machine

states not initially known. In this context different clustering algorithms can be better suited for the different objectives. To recognize the type of production being carried on by the machine, the K-Means algorithm can benefit from the starting considerations on the number of production types. At the same time, a more advanced algorithm like HDBSCAN seems to provide better performances when identifying unknown machine conditions from sensors installed on the machine. Future works will involve the implementation of an online dashboard prompting the type of production in execution and the development of an anomaly detection algorithm based on the State Detection here presented.

REFERENCES

Bajer, M. (2017). Building an iot data hub with elasticsearch, logstash and kibana. *Proceedings - 2017 5th International Conference on Future Internet of Things and Cloud Workshops, W-FiCloud 2017*, 2017-January, 63–68.

Border, D. (2018). Programmable logic controllers and data traffic handling solutions. *ASEE Annual Conference and Exposition, Conference Proceedings*, 2018-June.

Campello, R.J., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, 160–172.

Cattaneo, L., Fumagalli, L., Macchi, M., and Negri, E. (2018). Clarifying data analytics concepts for industrial engineering. *IFAC-PapersOnLine*, 51(11), 820–825.

Commission, I.I.E. et al. (2016). Iec 62264-3.

Davis, R. (2016). An introduction to asset management. *Retrieved November*, 20, 2016.

Hastie, T., Friedman, J., and Tibshirani, R. (2001). *The elements of statistical learning*. Springer series in statistics New York.

Humby, C. (2006). Data is the new oil. *Proc. ANA Sr. Marketer's Summit. Evanston, IL, USA*.

Jardine, A., Lin, D., and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7), 1483–1510.

Kammoun, M. and Rezg, N. (2018). Toward the optimal selective maintenance for multi-component systems using observed failure: applied to the fms study case. *International Journal of Advanced Manufacturing Technology*, 96(1-4), 1093–1107.

Kusiak, A. and Verma, A. (2011). A data-mining approach to monitoring wind turbines. *IEEE Transactions on Sustainable Energy*, 3(1), 150–157.

Le Noac'h, P., Costan, A., and Bougé, L. (2017). A performance evaluation of apache kafka in support of big data streaming applications. *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, 2018-January, 4803–4806.

Lin, S.W., Miller, B., Durand, J., Joshi, R., Didier, P., Chigani, A., Torenbeek, R., Duggal, D., Martin, R., Bleakley, G., et al. (2015). Industrial internet reference architecture. *Industrial Internet Consortium (IIC), Tech. Rep.*

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 14, 281–297.

Mahnke, W., Leitner, S.H., and Damm, M. (2009). *OPC unified architecture*. Springer, Berlin, Heidelberg.

Malzer, C. and Baum, M. (2020). A hybrid approach to hierarchical density-based cluster selection. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 223–228.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Hung Byers, A., et al. (2011). *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute.

Mattioli, J., Perico, P., and Robic, P.O. (2020). Artificial intelligence based asset management. *SOSE 2020 - IEEE 15th International Conference of System of Systems Engineering, Proceedings*, 151–156.

McInnes, L. and Healy, J. (2017). Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 33–42.

McInnes, L., Healy, J., and Astels, S. (2017). hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11), 205.

Mittal, M., Goyal, L.M., Hemanth, D.J., and Sethi, J.K. (2019). Clustering approaches for high-dimensional databases: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3), e1300.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Quadrini, W., Galparoli, S., Nucera, D., Fumagalli, L., and Negri, E. (2021). Architecture for data acquisition in research and teaching laboratories. volume 180, 833–842. doi:10.1016/j.procs.2021.01.333.

Raheja, D., Llinas, J., Nagi, R., and Romanowski, C. (2006). Data fusion/data mining-based architecture for condition-based maintenance. *International Journal of Production Research*, 44(14), 2869–2887.

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.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., Er, M.J., Ding, W., and Lin, C.T. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267, 664–681.

Strachan, S.M., McArthur, S.D., Stephen, B., McDonald, J.R., and Campbell, A. (2006). Providing decision support for the condition-based maintenance of circuit breakers through data mining of trip coil current signatures. *IEEE Transactions on Power Delivery*, 22(1), 178–186.

Williams, T.J. (1994). The purdue enterprise reference architecture. *Computers in industry*, 24(2-3), 141–158.

Zezulka, F., Marcon, P., Bradac, Z., Arm, J., Benesl, T., and Vesely, I. (2018). Communication Systems for Industry 4.0 and the IIoT. *IFAC-PapersOnLine*.

Zhong, R., Xu, X., Klotz, E., and Newman, S. (2017). Intelligent manufacturing in the context of industry 4.0: A review. *Engineering*, 3(5), 616–630.