Dylan Molinié, Kurosh Madani and Véronique Amarger

LISSI Laboratory EA 3956, University of Paris-Est Créteil, Sénart-FB Institute of Technology, Campus of Sénart, 36-37 Rue Georges Charpak, F-77567 Lieusaint, France

dylan.molinie@gmx.fr; madani@u-pec.fr; amarger@u-pec.fr

was originally published in the book

Advances in Computational Intelligence

(Print ISBN: 978-3-031-43077-0 / Online ISBN: 978-3-031-43078-7)

and published by

Springer Nature Switzerland AG

This paper was originally presented in the

17th International Work-Conference on Artificial Neural Networks (IWANN 2023)

held in

Ponta Delgada, Azores, Portugal
19-21 June 2023

This manuscript is related to the project

Hyperconnected architecture for high cognitive production plants HyperCOG

which received funding from the

European Union's Horizon 2020 research and innovation program

under grant agreement

No 869886

with persistent identifier

HyperCOG-869886

https://www.hypercog.eu/

This preprint has not undergone peer review (when applicable) or any post-submission improvements or corrections. The Version of Record of this contribution is published in *IWANN2023: Advances in Computational Intelligence*, and is available online at:

DOI 10.1007/978-3-031-43078-7_36
URL https://link.springer.com/chapter/10.1007/978-3-031-43078-7_36
Event http://iwann.uma.es
Date 01 October 2023

Unsupervised Clustering at the Service of Automatic Anomaly Detection in Industry 4.0*

Dylan Molinié[0000−0002−6499−3959], Kurosh Madani, and Véronique Amarger

LISSI Laboratory EA 3956,
Université Paris-Est Créteil, Sénart-FB Institute of Technology,
Campus de Sénart, 36-37 Rue Georges Charpak, F-77567 Lieusaint, France
{firstname,lastname}@u-pec.fr

Abstract. Industrial processes are among the most complex systems, for they are dynamic, nonlinear and comprise many interdependent parts. In the scope of the contemporaneous fourth industrial revolution, known as Industry 4.0, the trend is to integrate Artificial Intelligence and hyper-connectivity to intelligently exploit any available system's resources. A key issue for system management is the control of anomalies, which may cause severe failures if not corrected rapidly, or affect product quality; defining and knowing how to handle them is thus of major importance. In this paper, we propose to apply Machine Learning-based unsupervised clustering to industrial data to automatically identify the anomalies historically encountered; this is expected to help understand the system and define a framework for diagnosis of failures. We show that unsupervised clustering is able to detect salient groups of data, which can therefore be classified as anomalies by comparing them to the regular system's behaviors, obtained using another round of unsupervised clustering.

Keywords: Machine Learning · Industry 4.0 · Automatic diagnosis · Anomaly detection · Unsupervised clustering · Data Mining.

1 Introduction

With the global trend of production optimization and resource usage reduction, the industrial sector is mutating in depth: more and more projects emerge to propose new, innovative ways to handle these keys issues [14]. To that purpose, one may consider two fashions to operate [13]: 1) Always keep a vigilant eye on the systems to master them and control how they evolve; 2) Build predictive, automatic models to estimate their future evolution and react accordingly.

The first point corresponds to what is often done in most industries: senior, highly experienced technicians regularly check the state of the processes, and mechanically operate the actuators so as to keep them in their operating areas. Constant monitoring is commonly used, but it is expensive, and when a senior operator leaves or retires, his or her knowledge does as well, knowledge which may

* This paper has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 869886 (project HyperCOG).

be hard to replace. Likewise, due to systems' intrinsic complexity, a small change in a process may get worse if not handled properly, and cause severe failures by snowball effect; unfortunately, such small changes might not be visible before it is too late to prevent them from affecting the other processes, which can thus only be handled afterwards, when their negative effects become noticeable [4].

Although real-time analysis by an expert is the actual "standard" way to monitor industrial systems, new possibilities emerge with the development of the Artificial Intelligence sector [1]. Nowadays, the industrial sector is turning to the integration of Machine Learning, and Artificial Intelligence more generally, alongside highly connected systems. Such combination provides the basis of a new industrial revolution, known as Industry 4.0, which aims to globally improve system management, by better exploiting the production units, while proposing smart, cognition-oriented tools to handle and control them [5].

In that scope, a common problem one may encounter with any industrial system is process drift, i.e. when it evolves toward an unwanted direction [18]. Such events are detrimental to the industrial efficiency and product quality, and, as such, operators strive to prevent them from occurring, or, if not possible, strive to correct them as soon as possible. They are aptly assimilated to anomalies, in comparison to the normal way a system was crafted to behave; identifying, handling and correcting these anomalies is a cornerstone of system management, quality assessment, and is a key to help improve production chains [15].

There might be several ways to define and point out such anomalies, but intuition would assimilate them as any system drift, i.e. when any sensor reaches an unwanted range of values. Assuming a system behaves "normally" most of the time (i.e. in the way it was crafted for), most of its sensors' states (data) should take very similar values, contained within a restricted interval, whilst the abnormal events should not. Consequently, one may identify and isolate them by searching for salient regions in the feature space, where some data would form local, compact groups, apart from the largest ones (the regular "behaviors").

In the context of anomaly detection and assisted diagnosis, the first step consists in defining the different anomalies the system may encounter; to that purpose, assuming they form local groups of data, isolated from one another, we propose to use data-driven unsupervised clustering to isolate and classify the anomalies a system may encounter. Indeed, these Machine Learning-based algorithms aim to automatically regroup data sharing similarities into compact groups. With that definition of anomaly, unsupervised clustering seems to be the most appropriate way to automatically isolate and point out the events of a system [11]. Then, we propose to use higher level unsupervised clustering to identify the behaviors of the system (ground truth), to which the clusters can be compared to so as to classify them as either an anomaly or a true behavior.

We will begin this study by an overview of the already-existing techniques for anomaly detection, then we will introduce with more details the concept of unsupervised clustering and our proposed method for anomaly detection, and we will apply it to real industrial data, collected within the European project HyperCOG (introduced in section 4), and we will finally conclude this paper.

2 State-of-the-Art

When dealing with dynamic processes, such as industrial ones, fault diagnosis is of major importance, for an error should be corrected before it propagates and affects the other systems. An "anomaly" can either refer to system's fault or abnormal product's attribute; for the latter, for instance, a micro-crack on a telescope lens is such an anomaly; for the former, when a process gets out of its nominal range, this may impact product quality, or even the whole system.

With dynamic processes, the notion of anomaly can therefore be reduced as a state when some sensors take abnormal values, outside their nominal ranges. Such event can occur when a system is wrongly controlled, or even just by itself, especially with chemical compounds which may react differently than expected, due to the inertia of the chemical reactions. That last example illustrates what is known as a drift, i.e. when a process takes an unwanted direction for any reason; drifts are very common with industrial, dynamic systems. Such drifts may be the commonest sort of anomalies one may encounter with dynamic systems.

With that definition, the simplest way to identify the anomalies encountered by a system is certainly by comparing them to its regular behaviors (that one may call modes): an anomaly is therefore whatever differs from them.

In that scope, [3] proposed to use a Petri Net to build a dynamic model, which learns by example through time. Starting from a manual net, the model adds the different states the system passes through over time as new nodes. Although relevant, this approach relies on a manual initialization, which may be case-dependent, and the learning stage is ambiguous, since it does not really propose a way to distinguish between regular behaviors and abnormal anomalies.

In the same vein, [16] proposed HyBUTLA, which trades the Petri Net of [3] for a more classic finite-state automaton, but extended to the domain of hybrid systems (i.e. containing both analog and digital processes).

The main drawback of the approaches based on models is that it is generally difficult to identify the regular modes (behaviors) of a system, and to isolate the anomalies then. As a consequence, [2] proposed to apply unsupervised clustering to classify the data of a sensor through time, and then apply more classic statistical distribution functions to propose a model to the sensor, which can then be used to follow its temporal evolution. This tool was later integrated to a dedicated interface, named the SCADA method [10].

Some works preferred using Deep Learning for anomaly detection [15,17], but, as usual in that domain, it requires many labeled and classified examples for the training, which is prohibitive in an unknown context such as ours.

Finally, a last mention on the work of Latham et al. is worth being made [8]: they proposed Machine Learning-based tools to pre-process data so as to make them ready for more complex, higher level modeling and anomaly diagnosis-oriented approaches, such as applying a CNN to classify time-series images for anomaly detection in a steel industry context [7].

The main drawback with all these works is that they generally rely on much knowledge, for instance an initial automaton of the system, or many manually-labeled data, making these approaches poorly appropriate for unknown systems,

in blind contexts where the purpose is to automatically detect and classify the anomalies a system may encounter. For that reason, we propose to use tools from the domain of Data Mining to address the problem of anomaly detection, especially in the context of industrial process' drifts, for they generally require very little, if not no, information on the system to perform.

3 Proposed Methodology

We propose to use unsupervised clustering to isolate the system's anomalies, especially its drifts. We operate in two steps: a first clustering of the historical data, and a classification of these groups by comparing them to the regular system's behaviors, identified and delimited by using another round of clustering.

3.1 Clustering Algorithms

Clustering consists in gathering data into compact and homogeneous groups within which they share similarities, but not between the different groups. By considering a dissimilarity score, two data of a same cluster should have a low score, whilst two data belonging to two different clusters should have a high one. The purpose of clustering is essentially to find a partitioning of the feature space minimizing the dissimilarity scores between the data of a same cluster, for every cluster, whilst also maximizing these scores between the data of different clusters. From that perspective, clustering is mostly a matter of optimization.

There are plenty of clustering algorithms; they can be categorized into two categories: the supervised and the unsupervised ones. The former require the labels of the classes, i.e. the data must be labeled upstream by any means, and the purpose is to find the best borders delimiting the classes. The latter do no require such knowledge, they perform very generally, assuming nothing on the data or classes; they are more generic, but are also more random, for they draw their own conclusions, without possibility to confirm them. Since we operate in a blind, Data Mining-oriented context, only the second category can be considered.

One of the simplest unsupervised clustering is the K-Means [9], a divisive algorithm which draws K points, assimilates them as the clusters' barycenters, labels any data with the same class as that of the closest barycenter, updates the barycenters as the means of the so-built clusters, and redoes that procedure until satisfying any criterion, typically when few enough data change of category from a round to the next. The K-Means is one of the most widespread algorithms due to its simplicity: although naive, it is intuitive. Generally sufficient to get an idea on the regions of the feature space, it generally provides weak partitionings, highly sensitive to initialization, and is only able to propose linear borders.

Based on a similar idea, the Self-Organizing Maps (SOMs) replace the notion of mean of the K-Means by the concept of attraction [6]. Linked within a grid (referred to as "map" here), the barycenters are attracted by the samples: when a training data is drawn, the closest the barycenter, the more attracted it is. This attraction takes the shape of an update of position, which is propagated within

the whole grid, but which diminishes as the further away the barycenters get. Doing so allows to greatly speed up the learning stage, since every data updates the whole grid, and not only one cluster; moreover, by linking the barycenters to one another, their final topology respects that of the database. Also, a great advantage of the SOMs is that they are able to deal with even nonlinearly separable datasets, by using nonlinear learning functions (typically Gaussian-like).

Since the purpose of this paper is to point out the anomalies (drifts) of dynamic, industrial processes, assuming no previous information, the K-Means and the SOMs are two good candidates for that task. As a consequence, we will consider using both for a simple, first version of anomaly detection round.

3.2 Clustering for Behavior Identification and Anomaly Detection

The second step of our methodology consists in classifying the clusters provided by the K-Means or the SOMs; indeed, these algorithms partition the whole database, and there is no distinction between the clusters issued: they can either be regular states or true anomalies. Therefore, a way to distinguish between both is needed; often done manually, we propose a way to automatize the process, by comparing the clusters to the regular behaviors (modes) of the system.

We investigated the domain of behavioral identification in some of our previous works [11,13], which resulted in proposing a clustering method based on the SOMs, the Bi-Level Self-Organizing Maps (BSOMs), specifically crafted so as to automatically identify the behaviors of an unknown system, only using its historical sensors' data [12]. It relies on a two-step clustering: first, the dataset is clustered N times, thus providing several, possibly different partitionings, which are then all projected onto a new SOM.

In [12], we showed that this method achieves more accurate results than both the K-Means and the SOMs in the identification of the behaviors of an unknown system. As a consequence, we propose in this paper to use this approach to identify the regular behaviors of industrial systems, to which can be compared the clusters obtained in the first stage to automatically classify them.

3.3 Hybridization of Clustering for Anomaly Classification

Since the BSOMs are good in identifying the behaviors of an unknown industrial system, by using its historical data only, they can be used to create a reference, similar to a ground truth, of the system, using a somewhat statistical approach.

Meanwhile, since the manifestation of an anomaly is assumed to be a local, compact group of data, standing out from the others, it should be quite easily isolated by more classic clustering, such as the K-Means or SOMs. Unfortunately, these algorithms cannot distinguish between the nature of the groups (anomaly or behavior), nor can quantifiers: for instance, both an anomaly and a regular behavior should be compact, but the latter may comprise more data, whence the use of a statistical, averaged clustering algorithm such as BSOM.

Finally, to automatically label the clusters, and automatically point out the anomalies, knowledge greatly useful for system management and diagnosis, we

propose to compare the erratic clusters (K-Means/SOMs) to the averaged ones (BSOM): if a cluster is the middle of a behavior, then it is actually a part of it; else, it is an anomaly. Our methodology is depicted on Fig. 1.

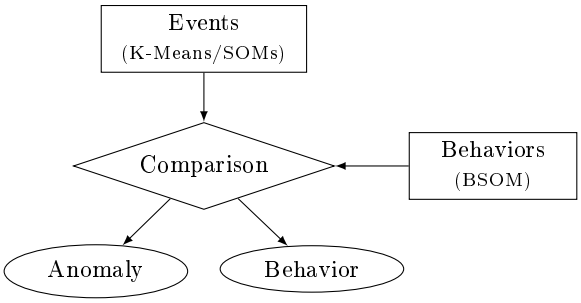

Fig. 1. The two-level methodology for anomaly classification.

4 Results

4.1 Presentation of the Dataset

This paper takes place in the context of the European project HyperCOG, which gathers fourteen partners, among which three industrial use-cases for validation. The purpose of this project is to study the feasibility of the hyper-connected cognitive plant with an Industry 4.0 orientation: a Cyber-Physical System is designed, within which intelligent, Machine Learning-based and cognition-oriented tools are integrated for automatic intelligent control and system management.

One of the industrial partners of that project is a cement plant situated in Turkey, belonging to the group ÇimSA. They produce white Portland cement and want to integrate high-level tools based on Artificial Intelligence, so as to ameliorate their management of their processes and smooth their production by identifying, predicting and eventually correcting the drifts of their systems.

In that context, they provided us with a scenarized dataset containing some real "events" (assimilated to anomalies). It consists of a six-day long dataset, from September 09, 2022 at 00:00:05 to September 15, 2022 at 00:00:00, at 5 minute intervals, for a total of 1440 data; they dispose of many sensors, but they provided eighteen key sensors for conciseness. In this study, for confidentiality concerns, the sensors' tags and units will not be shared; this does not impact the results, since the tools do not require that pieces of information to perform. Moreover, for the same reason, the data will be normalized between 0 and 1.

The events of this dataset are all the periods when any expert took some action to handle a key sensor being out of range; this led to four labeled events, plus an additional one, when a key sensor was out of range, but with no specific

action taken. The full dataset is depicted on Fig. 2, where every sensor is represented against time. The regular data are represented as blue crosses, whilst the events are highlighted as contiguous colored crosses, as mentioned in Table 1. For every column, the plots share the same abscissa axis (representing the time in minutes), and for every row, the plots share the same ordinate axis (representing the normalized data, with arbitrary unit (a.u.) and no dimension).

Although every of the eighteen sensors are represented here to be exhaustive, in all the following, we will only depict the Sensors 5, 12 and 14, for the events are easier to distinguish on them. Notice that of the sensors will be used for the training of the clustering algorithms nonetheless.

4.2 Anomaly Detection with Unsupervised Clustering

The first stage of our methodology is to apply unsupervised clustering, such as the K-Means (KM) or the Self-Organizing Maps (SOMs), to a scenarized dataset, i.e. containing real events which occurred during the recording. Before striving to classify these events, they must be identified; the purpose of this stage is therefore to check if unsupervised clustering is able to isolate the true anomalies from the rest of the data: the database will be split, but the question is to know if the clusters mean anything. Indeed, a dataset may be split in many fashions, but the ideal case would be to get the anomalies in unique and isolated clusters, whilst having all the non-event data in another cluster; nonetheless, due to the possible modes of a system, the "regular" data, it is unlikely that they form a unique group, whence the investigation of behavioral identification in next stage.

The database is clustered using both the K-Means and the SOMs, using a large number of clusters, to split it into many pieces; indeed, since there might

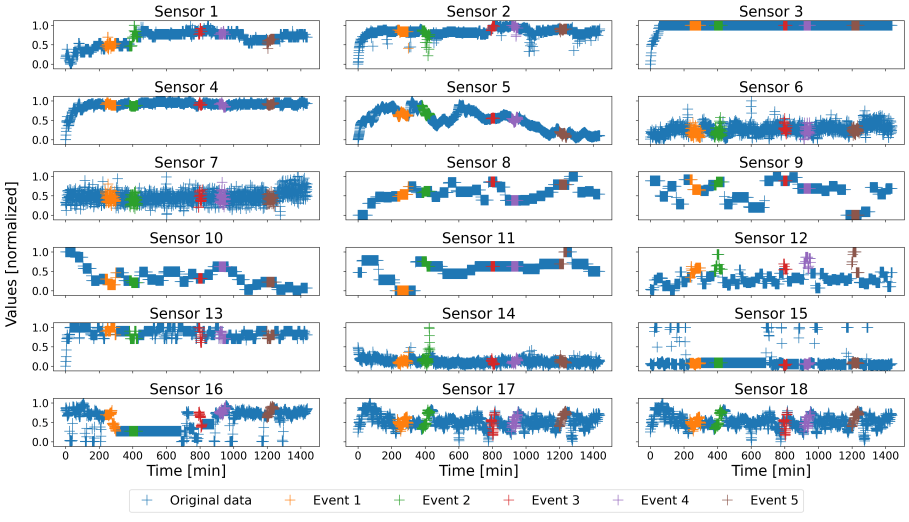

Fig. 2. The eighteen sensors and the corresponding events (all colors but blue).

Table 1. List of the events to be identified.

Events	Starting Time	Ending Time	Color Scheme
Event 1	2022-09-10 19:40:00	2022-09-11 01:00:00	Orange
Event 2	2022-09-11 07:45:00	2022-09-11 11:45:00	Green
Event 3	2022-09-12 18:00:00	2022-09-12 19:50:00	Red
Event 4	2022-09-13 04:10:00	2022-09-13 07:25:00	Purple
Event 5	2022-09-14 03:50:00	2022-09-14 07:35:00	Brown

be several events, which often comprise few data, it may result in many salient groups, thus there must be enough objective classes in the unsupervised clustering stage to represent them all. As a consequence, the K-Means will be trained with $K = 15$ and the SOM will consist of a 4×4 nodes grid. The training is made using any of the eighteen sensors of the database; the three selected in section 4.1 are depicted on Fig. 3, where the original data are on the uppermost row of plots, with the events represented as colored crosses (with respect to Table 1), and the clustered database using the K-Means and a SOM are represented on the middle and lowermost rows, respectively, where every cluster is represented by a unique color. Notice that the colors are arbitrary, for there can be no correspondence between the clusters issued by the K-Means and that of the SOM.

The clusters issued by both methods generally achieved to isolate every event but Event 3 (red), even though some inertia can be noticed. For instance, Event 1 (orange in the uppermost row of Fig. 3) can be assimilated to the cluster in cyan of the K-Means (middle row), and to that in brown of the SOM (lowermost row); in both cases the clusters are a little larger than the real event's data, but the region is nevertheless correctly identified. For both methods, the correspondences between their respective clusters and the real events are summarized within Table 2; globally, all the events were correctly identified, except Event 3, which is too small and which does not comprise truly salient data, which was missed.

4.3 Behavioral Identification with BSOM

In order to classify the clusters obtained in section 4.2, they can be compared to a ground truth; that piece of information is generally hard to obtain though,

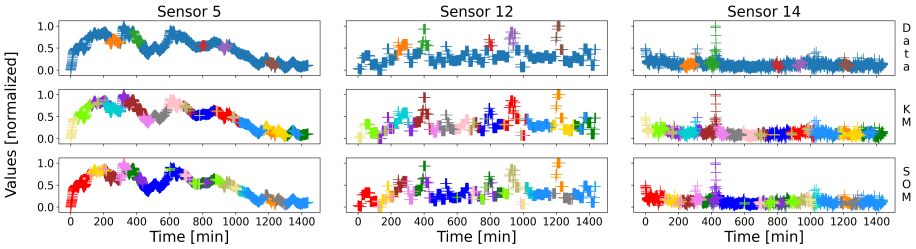

Fig. 3. The three selected sensors clustered using the K-Means and the SOMs.

Table 2. Correspondence between the clusters and the events

Events	Color Schemes		
	Database	K-Means	SOMs
Event 1	Orange	Cyan	Brown
Event 2	Green	Brown	Dark Green
Event 3	Red	Missed	Missed
Event 4	Purple	Red	Khaki
Event 5	Brown	Orange	Orange

thus, with respect to [13] and [12], we used a BSOM to cluster the database, considering all its data, so as to identify and delineate the processes' behaviors. The identified system's behaviors (modes) are depicted on Fig. 4, obtained using a set of 10 SOMs of size 3×3. The BSOM comprises six nonempty clusters, which can be assimilated to the historical behaviors of the system; actually, the blue, red, green and orange ones are true modes, but the brown one is more a catch-all cluster (somehow outliers), and the purple one is a real anomaly, which was, ironically, isolated by the BSOM also, due to the great saliency of its data.

4.4 Comparison of the Clusters with the Behaviors

Now that the database has been split into pieces by unsupervised clustering, isolating most of the anomalies as distinct groups, and that the behaviors of the system have been pointed out, both sets of results can be used to classify the clusters obtained in the first stage. Indeed, the motivation behind using an averaging clustering as of BSOM for behavioral identification is to get the typical state vectors corresponding to the different modes of the system; indeed, despite the presence of anomalies, one can assume that they are greatly rarer than the regular values, thus, using clustering allows to point out the main regions of the system (its standard ways to behave), and using an averaging clustering diminishes the impact of the outliers (the anomalies); as a consequence, the clusters' barycenters (also called patterns) should be representative of these modes.

In that context, in order to classify the clusters issued in the first stage, they can be compared to the system's behaviors, issued in the second stage. A relevant fashion to do so is to compare their respective patterns, for they are representative of the clusters, but with the impact of the outliers diminished.

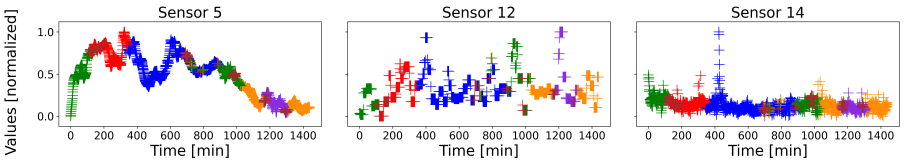

Fig. 4. The system's behaviors on the three selected sensors.

As a consequence, for a given cluster, the first step consists in searching for the closest behavior, by computing the Euclidean distances between its pattern and that of all the behaviors, and then selecting that with the lowest distance. Once identified, the pattern of this behavior can be compared to that of the considered cluster to be classified; this comparison can be a vectorial ratio, which indicates how close the cluster's barycenter is from the corresponding behavior's barycenter. Eventually, since an anomaly generally occurs for only a few sensors, the maximal value among all the dimensions can be considered: if high enough, it means that the cluster greatly differs from its closest behavior, thus it can be classified as an anomaly. This methodology is applied to all the clusters of both clustering methods, whose results are gathered within Table 3.

From this table, it is noticeable that the clusters corresponding to the anomalies (clusters 2, 4, 5 and 9 for the K-Means, and clusters 1, 2, 6 and 9 for the SOM, with respect to Table 2) got globally higher ratios than the others. For instance, for the K-Means, they got the highest ratios, meaning they are the most dissimilar clusters from their respective closest behavior: they are true anomalies. This statement is a little less true for the SOM: three of these four clusters got high dissimilarity scores, but the last one got quite a low; this is due to the fact that the dark green cluster issued by the SOM spans across two regions of the feature space (corresponding actually to the Events 2 and 3), thus it becomes similar to a true mode of the system, whence the proximity to the green behavior. For the SOM again, some clusters got high scores, even though they are not expected to be true anomalies, such as cluster 8: this is due to Sensor 15, whose values are very high for this cluster, which may be a regular way to behave and thus which was not tagged as an anomaly by the CimSA's experts (for it is actually not).

Table 3. Ratios between the clusters' patterns and the behaviors' patterns. The ratios of the clusters corresponding to the events (cf. Table 2) are highlighted in bold.

Clusters	K-Means		SOM	
	Closest Mode	Patterns' ratio	Closest Mode	Patterns' ratio
Cluster 1	1	16.73	1	12.04
Cluster 2	5	**23.94**	5	**23.21**
Cluster 3	2	13.76	1	**13.46**
Cluster 4	1	**23.87**	3	13.63
Cluster 5	4	**19.58**	1	15.13
Cluster 6	1	17.33	4	**21.52**
Cluster 7	1	13.02	6	13.18
Cluster 8	1	11.80	5	27.79
Cluster 9	6	12.73	1	**23.57**
Cluster 10	4	**21.85**	2	19.54
Cluster 11	5	11.91	4	33.17
Cluster 12	3	13.28	1	18.43
Cluster 13	1	12.64	4	13.34
Cluster 14	2	13.73	2	16.50
Cluster 15	3	14.98	1	19.03

5 Conclusion

In this paper, we proposed an automatic, Machine-Learning-based and data-driven methodology to address the problem of anomaly detection and identification of an unknown system, in an Industry 4.0 context. To that purpose, we proposed to proceed in two steps: first, partition the feature space of the system using a standard unsupervised clustering method (e.g., K-Means or Self-Organizing Maps) so as to split it into pieces and isolate the anomalies; then, partition it anew using a higher level, averaging clustering algorithm (e.g., BSOM) so as to point out the behaviors of the system, which can serve as ground truth; third, compare the clusters issued by the first stage to the behaviors issued by the second stage, so as to classify the clusters as anomalies or behaviors.

Applied to real industrial data, provided by a cement plant (CimSA group), as of a scenarized dataset comprising a handful of events which occurred during the recording and which led to the intervention of any experts to correct the related drifts, we showed that unsupervised clustering was globally able to isolate the event's data as distinct groups, that the averaging clustering proved able to point out a good estimate of the true modes of the system, and that comparing the clusters to these behaviors allowed to classify most of them as true anomalies.

Although promising, the main weakness of this study is the relative smallness of the dataset: it comprises 1440 samples only, which is great to give a first insight, but which may not be as representative as one would expect. Consequently, as future work, we are currently collecting more data and refining the notion of "events" and "anomalies" with our industrial partners to build larger, more representative and more meaningful datasets for anomaly detection. Moreover, we plan to refine our approach by applying intelligent modeling, which we expect will help follow the evolution of the system and prevent the anomalies.

References

1. Abdallah, M., Joung, B.G., Lee, W.J., Mousoulis, C., Raghunathan, N., Shakouri, A., Sutherland, J.W., Bagchi, S.: Anomaly detection and inter-sensor transfer learning on smart manufacturing datasets. Sensors **23**(1) (2023). https://doi.org/10.3390/s23010486
2. Calvo-Bascones, P., Sanz-Bobi, M.A., Welte, T.M.: Anomaly detection method based on the deep knowledge behind behavior patterns in industrial components. application to a hydropower plant. Computers in Industry **125**, 103376 (2021). https://doi.org/10.1016/j.compind.2020.103376
3. Dotoli, M., Pia Fanti, M., Mangini, A.M., Ukovich, W.: Identification of the unobservable behaviour of industrial automation systems by petri nets. Control Engineering Practice **19**(9), 958–966 (2011). https://doi.org/10.1016/j.conengprac.2010.09.004, special Section: DCDS'09 – The 2nd IFAC Workshop on Dependable Control of Discrete Systems
4. Gupta, V., Mitra, R., Koenig, F., Kumar, M., Tiwari, M.K.: Predictive maintenance of baggage handling conveyors using iot. Computers & Industrial Engineering **177**, 109033 (2023). https://doi.org/https://doi.org/10.1016/j.cie.2023.109033

5. Huertos, F.J., Masenlle, M., Chicote, B., Ayuso, M.: Hyperconnected architecture for high cognitive production plants. Procedia CIRP **104**, 1692–1697 (2021). https://doi.org/10.1016/j.procir.2021.11.285, 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0
6. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics **43**(1), 59–69 (1982). https://doi.org/10.1007/BF00337288
7. Latham, S., Giannetti, C.: Pre-trained cnn for classification of time series images of anti-necking control in a hot strip mill. In: The 9th IIAE International Conference on Industrial Engineering 2021 (ICIAE2021). pp. 77–84 (2021)
8. Latham, S., Giannetti, C.: Root cause classification of temperature-related failure modes in a hot strip mill. In: Proceedings of the 3rd International Conference on Innovative Intelligent Industrial Production and Logistics - IN4PL,. pp. 36–45. INSTICC, SciTePress (2022). https://doi.org/10.5220/0011380300003329
9. Lloyd, S.P.: Least squares quantization in pcm. IEEE Trans. Inf. Theory **28**, 129–136 (1982)
10. Maseda, F.J., López, I., Martija, I., Alkorta, P., Garrido, A.J., Garrido, I.: Sensors data analysis in supervisory control and data acquisition (scada) systems to foresee failures with an undetermined origin. Sensors **21**(8) (2021). https://doi.org/10.3390/s21082762
11. Molinié, D., Madani, K., Amarger, C.: Identifying the behaviors of an industrial plant: Application to industry 4.0. In: Proceedings of the 11th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). vol. 2, pp. 802–807 (September 2021). https://doi.org/10.1109/IDAACS53288.2021.9661018
12. Molinié, D., Madani, K.: Bsom: A two-level clustering method based on the efficient self-organizing maps. In: 2022 International Conference on Control, Automation and Diagnosis (ICCAD). pp. 1–6 (2022). https://doi.org/10.1109/ICCAD55197.2022.9853931
13. Molinié, D., Madani, K., Amarger, V.: Clustering at the disposal of industry 4.0: Automatic extraction of plant behaviors. Sensors **22**(8) (April 2022). https://doi.org/10.3390/s22082939
14. Pozzi, R., Rossi, T., Secchi, R.: Industry 4.0 technologies: critical success factors for implementation and improvements in manufacturing companies. Production Planning & Control **34**(2), 139–158 (2023). https://doi.org/10.1080/09537287.2021.1891481
15. Ruiz-Moreno, S., Gallego, A.J., Sanchez, A.J., Camacho, E.F.: Deep learning-based fault detection and isolation in solar plants for highly dynamic days. In: 2022 International Conference on Control, Automation and Diagnosis (ICCAD) (2022). https://doi.org/10.1109/ICCAD55197.2022.9853987
16. Vodenčarević, A., Bürring, H.K., Niggemann, O., Maier, A.: Identifying behavior models for process plants. In: ETFA2011 (2011). https://doi.org/10.1109/ETFA.2011.6059080
17. Wang, H., Liu, X., Ma, L., Zhang, Y.: Anomaly detection for hydropower turbine unit based on variational modal decomposition and deep autoencoder. Energy Reports **7**, 938–946 (2021). https://doi.org/10.1016/j.egyr.2021.09.179, 2021 International Conference on Energy Engineering and Power Systems
18. Yeshchenko, A., Di Ciccio, C., Mendling, J., Polyvyanyy, A.: Comprehensive process drift detection with visual analytics. In: Laender, A.H.F., Pernici, B., Lim, E.P., de Oliveira, J.P.M. (eds.) Conceptual Modeling. pp. 119–135. Springer International Publishing, Cham (2019). https://doi.org/10.48550/arXiv.1907.06386