This manuscript entitled

BSOM: A Two-Level Clustering Method Based on the Efficient Self-Organizing Maps

written by

Dylan Molinié and Kurosh Madani
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

was originally presented in the

6th International Conference on Control, Automation and Diagnosis (ICCAD)

held in

Lisbon, Portugal
13-15 July 2022

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/

The proposed form of the manuscript is that accepted by the conference Program Committee after having passed the peer-reviewed process, but not the final published version.
Link to the original publication:

DOI	10.1109/ICCAD55197.2022.9853931
URL	https://ieeexplore.ieee.org/document/9853931
Event	https://www.iccad-conf.com/iccad2022/
Date	18 August 2022

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Horizon 2020
European Union Funding
for Research & Innovation

BSOM: A Two-Level Clustering Method Based on the Efficient Self-Organizing Maps

Dylan Molinié
LISSI Laboratory EA 3956
Senart-FB Institute of Technology – UPEC
Campus de Sénart, Lieusaint, France
dylan.molinie@gmx.fr

Kurosh Madani
LISSI Laboratory EA 3956
Senart-FB Institute of Technology – UPEC
Campus de Sénart, Lieusaint, France
madani@u-pec.fr

***Abstract*—At the very beginning of the Industry 4.0 era, automated systems and automatic knowledge conceptualization are becoming more and more essential. The ever faster, ever more resource-demanding processes are raising a pressing need for highly efficient handling of the systems. While the industries produce ever more, there is less and less time for back-up and quality assessment; a piece of solution may come along a real-time and automated monitoring, based on sensors' data so as to assess products' validity: this is the areas of Behavior Identification and Anomaly Detection. In this paper, we propose to use Machine Learning and data-driven clustering to automatically identify the real behaviors of a system, and therefore its possible anomalies. More than the methodology, we mostly propose a more stable clustering method based on the Self-Organizing Maps to achieve that purpose. We apply both methodology and that improved clustering method to real industrial data, and we show that it is more efficient, more stable and more relevant to dynamic systems.**

***Index Terms*—Industry 4.0, Behavior Identification, Anomaly Detection, Data Mining, Clustering, Self-Organizing Maps**

I. Introduction

When dealing with information, Data Mining is often a very useful preliminary task, albeit tricky: it aims to automatically extract some knowledge about them, while making no previous, expert-based assumption – or as little as possible.

Most data carry a tremendous amount of information; experts are tasked with making sense of them, by extracting this raw information, or by using these data within a wider, more elaborated context. Many algorithms only require the data's values to work: they do not need to understand them. On the contrary, some algorithms rely on the information the data may carry; for instance, this is the case of Behavior Identification and Anomaly Detection, where the data themselves do not matter, only what they represent does.

Nonetheless, a very problematic question remains: what knowledge may prove to be helpful and valuable? Intrinsically, when dealing with a real system, lots of information can be useful; for instance, if one is told that the system's data follow a Gaussian distribution, this knowledge can drive the choice of the algorithms. Unfortunately, such a valuable knowledge is hard to get (it often results from a manual, expensive study), and it is also highly specific to the use-case (low generalization capability). In this case, a highly generalizable and no case-specific knowledge conceptualization approach would be very useful and greatly valuable, whence the use of Data Mining in the area of knowledge conceptualization.

The aptly entitled Behavior Identification area aims to extract and characterize the behaviors a system can enter in: if its regular behaviors are well known, any differing behavior observed over time, and especially when real-time monitoring the system, is very likely an abnormal event, i.e., an anomaly. This approach is very useful in industrial contexts, where a real-time monitoring of the products on the production lines is of major importance to assess their quality.

A possible way to automatically identify the behaviors of a system is clustering, for it gathers the similar data in compact and homogeneous groups; indeed, assuming a periodic and real-time sampling of a system, the most recurrent data should be that of the regular states, since they occur most often. As a consequence, their data should be very numerous and form large, compact groups in the feature space; the more isolated data would very likely represent outliers, i.e., abnormal events.

Many clustering methods exist, based on both Expert Systems and Machine Learning. Some of our previous works [1], [2] consisted in assessing the validity of the methodology that clustering is well-suited for pointing out the regular behaviors of a real industrial system, and to identify some irregular, unforeseen and prohibitive events. This work was based on both the kernel K-Means and the efficient Self-Organizing Maps; in this present paper, we extend this work using an improved clustering and by applying it to an industrial context.

To that purpose, we will list some works related to Behavior Identification and Anomaly Detection, then we will introduce some clustering methods, our proposal for improvement and our methodology for assessment. Finally, we will test them on real industrial data, and we will conclude this paper.

II. State of the Art

Behavior Identification and Anomaly Detection are two sides of the same coin: they are highly linked to each other, for an anomaly can be seen as an irregular, temporary behavior. Some states are the standard ways in which the system can behave, whereas some others should be avoided, for they might

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

be prohibitive or even harmful to it. As such, it is important to identify both the regular and abnormal behaviors of a system, especially in an automated context such that of Industry 4.0.

Albeit aging, a relevant work is COMMAS [3]; the authors proposed a statistical study of a set of real electrical transformers so as to point out their similarities. To do so, they considered a subset of "healthy" (fault-free) transformers and interspersed their similar states by building and training a Hidden Markov Model upon them; used as a reference, any state differing from that of this model can be assimilated to an anomaly. This work has two major drawbacks however: it requires a strong hypothesis as of the isolation of totally fault-free transformers to build the model; and two systems may look alike, but they can differ from one another in depth due to their intrinsic physics (defects, operating points, etc.).

In the vein of that work, it is worth mentioning [4], which uses a mutating Petri Net. An initial Net is built – manually –, and is then updated so as to account for the observed states the system passed through over time; once the Petri Net trained, it is anew used as a reference to compare any observed event or state: any differing from the model is tagged as an anomaly. This approach is relevant, but requires an initial knowledge (the initial Petri Net), and an ambiguity remains: after how long time can the training be deemed as complete?

Alongside the Hidden Markov Model and the Petri Net paradigms, HyBUTLA [5] proposed to use a good old finite-state automaton to follow the temporal evolution of a hybrid system. Alike the two previous works, it relies on a strong hypothesis, in this case the full automaton of the system, which can be hard, long and expensive to build.

Although deemed trustful and reliable, Expert System-based methods are long and expensive to complete: any possible case must be thought, identified, characterized, and the action to be taken must be defined. Therefore, approaches based on automatic knowledge extraction have emerged in recent years.

In that context, [6] proposed to apply clustering to the data of a given sensor so as to gather its similarities through time. This first clustering was used as a coarse pretreatment, to rough out the data, by pointing out their patterns, i.e., the different barycenters of the sensor (its temporal behaviors). Once obtained, they refined the clusters by applying some classic Probability Density Distributions. These refined clusters were later used in [7] to monitor the sensor through time and to detect its possible anomalies. Based on these works, a tool for control and data acquisition was later developed to serve as a demonstrator in Industry 4.0, the SCADA method [8].

In parallel with the works of Calvo-Bascones et al., in [1], [2] and [9], we also proposed to use clustering, and especially the Self-Organizing Maps, for they are highly accurate and efficient, but to the whole system (all sensors), and to operate in the sensors' feature space, not only in that of a single sensor. Moreover, we did not rely on any knowledge such that of the Density Distributions, and we focused our efforts on automatic characterization, especially in the second paper. We also tried the K-Means and the Kernel K-Means, but the Self-Organizing Maps positively proved to be the most efficient and accurate.

III. Clustering and Methodology

A. Classical Clustering

Clustering consists in gathering the data sharing similarities; to do so, clustering algorithms statistically study a set of data, and aim to conceptualize their observations. A possible way is to estimate the different groups (labels) by semi-randomly drawing a set of barycenters, and then aggregate all the nearest (according to a distance) data around them. A step of learning is applied so as to make them move toward some positions which minimize some criteria; in general, the criterion is the minimization of the average intra-cluster distances, for all the clusters at the same time. By assimilating an average intra-cluster distance to an energy, it can be seen as the minimization of the energies of the subsystems (the clusters), while also minimizing the total sum of these local energies. As a consequence, clustering is mostly a matter of optimization.

Perhaps the most well-known algorithm is the K-Means [10]: it randomly draws a set of K data instances, aggregates the nearest data around them as full clusters, and then updates the barycenters as the empirical means of the so-built clusters; that procedure is repeated until satisfying some criterion. It works well with simple and linearly separable datasets, but achieves poor results with real, nonlinear data; the K-Means is a reference though and must be investigated anyway.

Alongside, an efficient solution to the above-mentioned problem is the Self-Organizing Maps SOMs [11]: they are resource-saving, fast and able to separate even nonlinear datasets by using a nonlinear learning function [9]. They act as a neighboring K-Means: the different clusters, called "nodes" here, are linked to each other within a grid, and the update of a node is propagated to the whole grid to fasten the learning.

The SOMs work with patterns rather than with means: a node is defined by a feature vector, which generally differs from the mean of the data contained within; the K initial patterns are drawn randomly, and then updated by an iterative learning procedure, executed with every data x of the database.

B. Two-Level Self-Organizing Map

Even though the SOMs work well, they might deserve some improvements. One of them is the usage of different learning functions; however, Gaussian-like expressions are generally used, which are perfectly suited to most databases. Another amelioration might be the mutation of the grid itself; nonetheless, a fixed size is generally sufficient for most cases, and a possible refinement is always possible, as tested and discussed in [1]; actually, such an evolution was already investigated and gave birth to the Neural Gas, which is somehow a SOM with a dynamic grid (possible pruning or addition of new nodes).

Due to their stochasticity, the main drawback of the SOMs is the impossibility to obtain the same result twice. Indeed, the draw of the initial patterns and of the data used for the map's update must be random, in order to avoid bias, but this also means that the final patterns will very likely slightly differ from a map to another; as a consequence, their clusters are almost always slightly different. The nodes are not numbered,

and even though the data's topology is conserved inside the SOM's grid, the nodes' order can be reversed or interchanged: it becomes very hard to keep track of the nodes, of the clusters and of the data themselves. As such, it is barely possible to blindly attest which map is the closest to reality; to be as realistic as possible, a consensus should be made on the maps.

A solution to that problem is to take advantage of the self-organization capability of the SOMs to diminish the clustering randomness; following that idea, we propose BSOM, a two-level map based on the SOMs, which essentially consists in projecting a set of maps on a new one. The idea is to create M maps, then constitute a new database with their respective K node's patterns, and eventually train a last grid upon it. Indeed, each map can be seen as somehow a pretreatment, a coarse possible clustering. Moreover, since several maps are fused together, the results should be more relevant and more consistent. The projection of several maps on a new one allows to take advantage of the benefits of randomness while getting a clustering more consistent and less mutating over the rounds. The operating principle of BSOM is depicted as of Fig. 1.

That idea is similar to the Multi-Layer Perceptron, since each node of the final grid is itself a full map. Whilst such a *mise en abyme* generally increases the accuracy of the method, a deeper stratification might lead to an overlearning which may highly hamper the generalization capability of the method.

Notice that setting different learning parameters to each map can have side beneficial effects, for it increases the clusterings heterogeneity; these meta-parameters are often set to very general values, and several configurations cover more cases, eventually merged by BSOM, which statistically features all of them. The grid's size can also be changed; nonetheless, a consistent number should be preferred, since comparing a 4-node map with a 90-node map makes no sense: they can in no way be similar, due to their intrinsic topology. Moreover, the subgroups obtained by the node's patterns should be as numerous as the first-level maps' size, and therefore a wider second-layer grid is not really relevant, nor even beneficial.

C. Map comparison

Beside the clustering improvement we proposed above, we also propose a way to compare two clusterings; indeed, there exists no method to strictly compare two sets of clusters. One might have recourse to some classical indicators, such as the cluster's Density [9], the Average Standard Deviation [12] or else the Silhouette Coefficients [13]; these metrics are highly useful to estimate the cluster's quality [1], [2], but less to statistically compare two sets of clusters. For this reason, we propose a new methodology, based on the "ground truth".

Assuming we have access to the ground truth, i.e., the real clusters of the database (its behaviors), it is possible to estimate how close each method has come to it by comparing the data of their clusters to the expected groups. To do so, we can study how each cluster has been distributed to the different behaviors and then compute some statistics to evaluate the relevancy of each method; Tab. I gives an example of such distribution – obtained with a true SOM applied to the data of section IV.

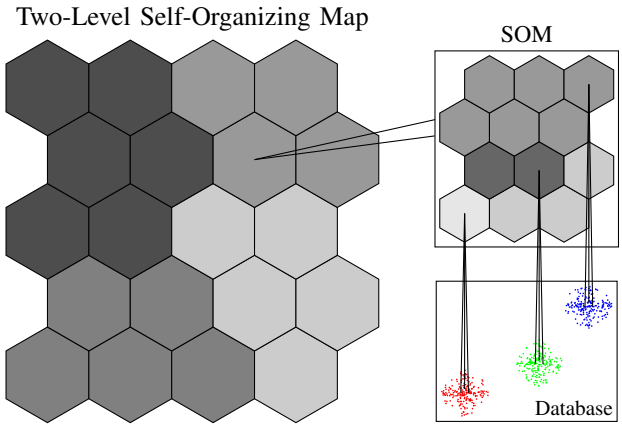

Fig. 1. BSOM operating principle.

One can observe up to four configurations. The first is the intersection of B1 with C1 and C2 (values in blue): they are representative of a behavior split into two clusters; in the following, such case will be referred to as "behavior split". Then, the intersection of C6 with B4 and B5 (in red) are characteristic of an overlap of behaviors, i.e., two distinct behaviors contained by a single cluster, which means that the clustering failed in separating them; this will be called a "behaviors overlap". Third comes the values at the crossroads of B2 with C4 and C5, plus B3 with C4 (in green): in this case, both events might be considered, i.e., a behavior split as of B2-C4 and B2-C5, but also a behaviors overlap with C4-B2 and C4-B3; this case is the worst, but generally rare and limited, since, as an example here, there are both events, but the intersection B2-C4 comprises few data. The fourth event is an almost empty cluster with C3, which can happen with the SOMs, meaning the grid might be a little too large, even though this is not a big problem since C3 leads to neither an overlap nor a split. Another event may appear as of an empty row, meaning a behavior would be missed by the clustering.

Due to the intrinsic stochastic characteristic of the SOMs, significant overlaps and/or splits are commonplace; as such, the question of the minimal number of data contained within an intersection to be taken into account should be considered. For instance, is the behavior split B2-C4-C5 a real split or only a simple limitation of the SOM? To answer that, we propose to consider only the intersections containing a certain part of the behavior, for instance 15%; in our case, B2 comprises

TABLE I
EXAMPLE OF BEHAVIORS' DATA DISTRIBUTION OVER THE CLUSTERS (BEHAVIOR X = "BX"; CLUSTER X = "CX")

	C1	C2	C3	C4	C5	C6	Total
B1	20,223	16,016	0	0	0	0	36,239
B2	0	0	0	766	23,401	1	24,168
B3	0	0	0	4761	0	0	4761
B4	0	0	0	0	0	170	170
B5	0	0	1	0	0	147	148
Total	20,223	16,016	1	5527	23,401	318	65,486

24,168 data, thus an intersection is considered if, and only if, it contains at least 3625 of them. Consequently, a behavior can be considered as split in as many clusters as there are intersections comprising at least 15% of its data, and can be considered as clearly and uniquely identified if there is a single intersection composed of at least 85% of its data.

This methodology works equally for an overlap: there is one only if a cluster intersects at least two behaviors, comprising at least 15% of each behavior's data, and there is no overlap if there is a single intersection comprising at least 85% of them.

To compare two sets of clusters, and thus two clustering methods, we propose to estimate the number of behavior splits and overlaps of both: that achieving the lowest number of occurrences can be assumed to be more trustful, since closer to the ground truth. Moreover, in case of a split, we compute the mean Euclidean distance between the clusters' patterns of the intersections involved in the split, as well as the mean Ward Linkage; these values are useful to measure the badness of the split, since even if a unique behavior has been cut into two clusters, they might be very close (closer from each other than from any other), and could therefore be merged later on. For the overlaps, only the Ward Linkage is usable, since the Euclidean distance is not applicable to a single cluster; this measure also estimates the seriousness of the overlap, and indicates if a further refinement could be possible.

Since this method is above all statistical, we must repeat it a sufficient number of times to ensure its representativeness; an estimation of that number n is the Cochran formula (1).

$$n = p(1-p)\left(\frac{Z}{m}\right)^2 \tag{1}$$

where m is the margin error, p is the proportion in the population with the desired characteristic, and Z is the standard score (Z score), defined through the confidence level.

By setting m to 0.03, Z to 1.96 (confidence level of 95%), and p to 0.5 (with m and Z set, $p = 0.5$ is the global maximum of n, thus the worst case), we yield $n \approx 1067$.

IV. Database and Results

In the context of HyperCOG, we collaborate with several industrialists to test our methods on real industrial data, with an Industry 4.0 perspective. One of these partners is Solvay®, a chemistry plant specialized in the extraction of Rare Earths. Note that we will not introduce further their data, nor their related sensors, for confidentiality concerns; this is not a real problem though, since we are assessing a blind unsupervised methodology. Two sensors of the database are represented as of the two leftmost columns of Fig. 2 to illustrate both the raw data and the real system's behaviors: the steady state B1 is in blue, the shutdown B2 in orange, the startup B3 in green, the power-off procedure B4 in violet, and the abnormal event B5 in red. We will apply the K-Means, SOM and BSOM to these data to show their benefits and limitations; these five behaviors, manually pointed out here, are the objective ones to which the clustering given by each method will be compared. Notice that BSOM will use 10 maps in the following, and K is set to 9.

Let us begin with a qualitative comparison of the methods; their respective clusters are depicted on the three rightmost columns of Fig. 2. As expected, BSOM achieved the best and most representative results, with less overlaps and splits than the two others. On the opposite, the K-Means unsurprisingly achieved the poorest results, with many overlaps, splits, and a low continuity in the clusters: it is difficult to say which cluster corresponds to which behavior, for the unity and isolation of the clusters is very blurry. Finally, SOM also got good results, with larger overlaps and slightly more scattered clusters than BSOM, but clearly better than the K-Means though. The fact that the K-Means achieved the worst results, then SOM and eventually BSOM is not surprising, for BSOM is an improvement of SOM, which is itself an improved K-Means.

All about SOM and BSOM, both proved their are able to correctly identify the system's major behaviors, but both failed in identifying the abnormal event. BSOM correctly identified the power-off, the shutdown and the startup; the steady state was correctly isolated, but distributed into two distinct clusters

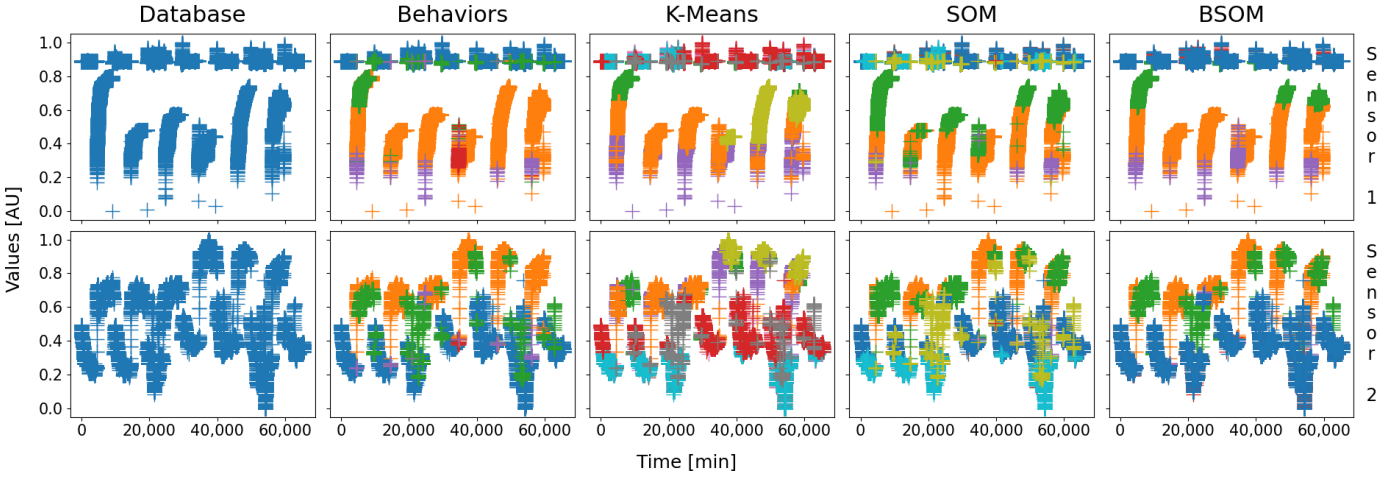

Fig. 2. Behavior's data examples – Among the eight sensors of the database, two are represented here, one per row. For each row, from the leftmost to the rightmost column: the raw data, the five reference behaviors (ground truth), the clustering given by the K-Means, SOM, and eventually BSOM.

(due to another sensor which can take two clear values in the steady state). Although not erroneous, SOM got more scattered results: the power-off was correctly isolated, but the startup was merged with a large part of the shutdown, and the steady state was split into three clusters; moreover, the light green cluster caught some data from the startup (green) and parts of the steady state (blue, cyan, plus barely visible brown).

Now that BSOM is qualitatively validated over the K-Means and SOM, it is time to assess the three methods quantitatively, by following the methodology introduced in section III-C. In order to conduct a statistical study, according to (1), we applied each clustering method 1067 times, and we enumerated the behavior splits and overlaps for each; the splits are gathered in Tab. II, and the overlaps in Tab. III.

Let us begin with the first table, i.e., the splits. For each of the five behaviors of reference (B1, B2, ...), the number of clusters within which their data were distributed is indicated as of row "Splits", which contains two sub-rows: the upper is the number of clusters containing at least 15% of the data of the corresponding behavior, whilst the lower is the number of occurrences over the 1067 maps. For example, with SOM, B1 was uniquely identified 0 time, was split into 2 clusters 77 times, and in 3 clusters 990 times. The "Euclid" line corresponds to the mean Euclidean distance between any couple of node's patterns when there are more than two clusters, and "Ward" is the mean Ward Linkage between the different clusters containing the same behavior.

Let us consider first the number of splits for each method. On the one hand, the K-Means very often split the behaviors into at least two behaviors (up to 4 for the largest ones), and almost always missed B4 and B5; on the other hand, both SOM and BSOM very often fully identified the behaviors, or split them into two clusters, but rarely more, with the exception of B1 for SOM, often cut in three. The K-Means clearly failed in identifying the behaviors as full entities; on the opposite, although not perfect, both SOM and BSOM proved that they are well-suited to the behavior identification we are dealing with, even if a further refinement is manifestly required.

Concerning the intrinsic results, the clusters of BSOM are clearly the most consistent and nearest to the real behaviors. The K-Means drowned the smallest behaviors into the largest ones and split them into many pieces; SOM and BSOM achieved similar results, nearer to the reality, but BSOM isolated (unique clusters) the behaviors more often, produced much less splits, and very rarely led to more than two clusters.

The last observation is about the quantifiers "Euclid" and "Ward": the first is globally better with the clusters obtained with BSOM than that obtained with the K-Means and SOM, meaning the sub-clusters of a given behavior are globally nearer from each other, and thus would be easier to merge. However, the Ward Linkage is often higher with BSOM, meaning the clusters are slightly more dissimilar. The data seem to have been more accurately separated from each other (clearer borders between the groups), which should be a good news, but in fact means that the clusters are more dissimilar, whilst they should not, for they belong to the same behavior. Consequently, the merging could be hardened if based on the Ward Linkage, whereas eased if based on the node's patterns.

As a conclusion, from the behavior split point of view, BSOM gave more consistent and more reliable clusters, nearer to the real behaviors than that of the K-Means and SOM.

Now, let us check it out with Tab. III, where the overlaps are represented in the first row, where "BX-BY-BZ" means that BX, BY and BZ are overlapping, the "Occurrences" row numbers how many times this overlap occurred over the 1067 maps, and "Ward Linkage" is anew the mean Ward Linkage between the intersections of the cluster with each behavior.

Firstly, except anecdotal B2-B5, B2-B3-B4 and B3-B4, the three methods led to the same overlaps, especially B2-B3, B4-B5 and B3-B4-B5. These three sets correspond to a merging of the different phases of the shutdown, which represent a unique behavior if one does not look for a deeper distinction.

Secondly, the K-Means globally achieved fewer overlaps, but only due to the fact that it missed and drowned B4 and B5 into larger clusters (cf. Tab. II); these two behaviors are very often involved in the observed overlaps, thus missing them

TABLE II
BEHAVIOR SPLITS OVER THE 1067 MAPS

		B1					B2				B3				B4	B5
K-Means	Splits	1 0	2 71	3 299	4 486	5 189	1 0	2 338	3 527	4 33	1 0	2 582	3 51	4 3	2 1	2 6
	Euclid	—	0.25	0.25	0.26	0.26	—	0.24	0.26	0.24	—	0.45	0.43	0.44	0.2	0.43
	Ward		467.2	318.1	279.0	249.0		334.3	268.3	185.7		565.5	184.2	124.4	131.5	499.8
		B1			B2			B3			B4			B5		
SOM	Splits	1 0	2 77	3 990	1 116	2 896	3 55	1 382	2 679	3 6	1 892	2 167	3 8	1 817	2 242	3 8
	Euclid	—	0.27	0.26	—	0.26	0.32	—	0.42	0.44	—	0.74	0.66	—	0.87	0.67
	Ward		589.7	378.4		372.0	439.3		404.8	449.6		237.2	302.9		107.0	103.1
		B1			B2			B3			B4			B5		
BSOM	Splits	1 363	2 704	3 0	1 973	2 94	3 0	1 504	2 559	3 4	1 901	2 163	3 3	1 979	2 86	3 2
	Euclid	—	0.24	—	—	0.28	—	—	0.40	0.32	—	0.55	0.54	—	0.45	0.33
	Ward		616.3			520.7			565.5	335.6		145.6	146.0		55.1	22.8

TABLE III
BEHAVIORS OVERLAPS OVER THE 1067 MAPS

		B2-B3	B2-B4	B4-B5	B2-B4-B5	B2-B3-B4-B5	B3-B4-B5	B2-B5	B2-B3-B4	B3-B4	B3-B5
KM	Occurrences	362	1	318	692	51	8	2	0	0	1
	Ward Linkage	63.20	4.95	2.03	46.54	131.54	60.94	25.26	—	—	52.85
		B2-B3	B2-B4	B4-B5	B2-B4-B5	B2-B3-B4-B5	B3-B4-B5	B2-B5	B2-B3-B4	B3-B4	B3-B5
SOM	Occurrences	656	297	530	269	19	5	17	107	23	5
	Ward Linkage	106.25	47.59	1.50	33.12	61.08	52.36	23.54	63.29	35.11	61.42
		B2-B3	B2-B4	B4-B5	B2-B4-B5	B2-B3-B4-B5	B3-B4-B5	B2-B5	B2-B3-B4	B3-B4	B3-B5
BSOM	Occurrences	551	89	924	39	4	3	0	44	14	0
	Ward Linkage	141.93	33.65	1.75	36.30	59.53	52.96	—	65.12	41.93	—

also wrongly means no overlap. Besides, regarding to SOM and BSOM, there are generally fewer bad overlaps with the second; that is mostly true for the largest behaviors though, but not for the smallest, for which a lower number of overlaps was achieved by SOM. This is due to the statistical approach of BSOM, which tends to avoid small clusters by merging them in larger ones, for several, scattered clusters can either be seen as distinct behaviors, or very close representatives of the same behavior. This statistical study is therefore more prone to merge the smallest clusters, but this merging is highly beneficial for wider clusters, such as B2-B3 for instance, whose number of overlaps has been greatly reduced.

Thirdly, the Ward Linkage also informs us that BSOM generally led to less dissimilar clusters; this could imply that the clusters might overlap, they might comprise more compact data, on the edge of the overlapping behaviors.

As a conclusion, after having tested the K-Means, SOM and BSOM on real industrial data, it is noticeable that the two last were globally capable to identify the system's major behaviors, but not the first one; nonetheless, BSOM globally achieved better results, with more compact and consistent clusters, nearer to the real, objective behaviors, and greatly reduced the splits. That being said, BSOM reduced the number of overlaps for the largest behaviors, but increased it for the smallest behaviors, which is an obvious limitation.

V. CONCLUSION

In this paper, we dealt with the tricky problem of the Behavior Identification in an industrial context, with an Industry 4.0 perspective; we proposed to use Data Mining to solve it, and especially clustering. To that purpose, we introduced the very well-known and highly efficient Self-Organizing Maps, but we also proposed an improvement to them, as of Two-level Self-Organizing Maps BSOM, which builds a set of maps, and then projects them on a new, final one. We also proposed a methodology to compare two set of clusters (clusterings), based on a statistical study of the splits and overlaps of the real behaviors of the system, using its ground truth as reference.

We applied the K-Means, SOM and BSOM to real industrial data and assessed them using our methodology. We showed that BSOM was more resilient, with more compact and consistent clusters, nearer to the ground truth, but also slightly increased the number of overlaps, due to a fusion of the closest behaviors as full ones. A possibility to compensate that could be to apply BSOM to the data, keep its largest clusters, and eventually refine its smallest ones by a single SOM.

Our future work will consist in using BSOM in a hierarchical clustering. Indeed, BSOM generally detected the main behaviors, but split or overlapped them; we strongly believe that some indicators and tests can blindly inform us on the actual defaults. We want to use these indicators in a split-and-merge fashion so as to refine the clustering, based on BSOM.

REFERENCES

[1] D. Molinié, K. Madani, and C. Amarger, "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*, vol. 2, pp. 802–807, September 2021.

[2] D. Molinié, K. Madani, and V. Amarger, "Clustering at the disposal of industry 4.0: Automatic extraction of plant behaviors," *Sensors*, vol. 22, April 2022.

[3] A. Brown, V. Catterson, M. Fox, D. Long, and S. McArthur, "Learning models of plant behavior for anomaly detection and condition monitoring," in *2007 International Conference on Intelligent Systems Applications to Power Systems, ISAP*, vol. 15, pp. 1 – 6, 12 2007.

[4] M. Dotoli, M. Pia Fanti, A. M. Mangini, and W. Ukovich, "Identification of the unobservable behaviour of industrial automation systems by petri nets," *Control Engineering Practice*, vol. 19, no. 9, pp. 958–966, 2011. Special Section: DCDS'09 – The 2nd IFAC Workshop on Dependable Control of Discrete Systems.

[5] A. Vodenčarević, H. K. Bürring, O. Niggemann, and A. Maier, "Identifying behavior models for process plants," in *ETFA2011*, 2011.

[6] P. Calvo-Bascones, M. A. Sanz-Bobi, and T. Álvarez Tejedo, "Method for condition characterization of industrial components by dynamic discovering of their pattern behaviour," in *ESREL2020*, November 2020.

[7] P. Calvo-Bascones, M. A. Sanz-Bobi, and T. M. Welte, "Anomaly detection method based on the deep knowledge behind behavior patterns in industrial components. application to a hydropower plant," *Computers in Industry*, vol. 125, p. 103376, 2021.

[8] F. J. Maseda, I. López, I. Martija, P. Alkorta, A. J. Garrido, and I. Garrido, "Sensors data analysis in supervisory control and data acquisition (scada) systems to foresee failures with an undetermined origin," *Sensors*, vol. 21, no. 8, 2021.

[9] D. Molinié and K. Madani, "Characterizing n-dimension data clusters: A density-based metric for compactness and homogeneity evaluation," in *Proceedings of the 2nd International Conference on Innovative Intelligent Industrial Production and Logistics – IN4PL*, vol. 1, pp. 13–24, INSTICC, SciTePress, October 2021.

[10] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, pp. 651–666, June 2010.

[11] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, Jan. 1982.

[12] M. Rybnik, *Contribution to the modelling and the exploitation of hybrid multiple neural networks systems : application to intelligent processing of information*. PhD thesis, University Paris-Est XII, France, Dec. 2004.

[13] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 11 1987.