Dylan Molinié, Kurosh Madani and Abdennasser Chebira

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; chebira@u-pec.fr

was originally published in the book

Innovative Intelligent Industrial Production and Logistics

(Print ISBN: 978-3-031-37227-8 / Online ISBN: 978-3-031-37228-5)

and published by

Springer Nature Switzerland AG

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 *Innovative Intelligent Industrial Production and Logistics*, and is available online at:

DOI 10.1007/978-3-031-37228-5_4
URL https://link.springer.com/chapter/10.1007/978-3-031-37228-5_4
Event https://link.springer.com
Date 07 July 2023

HyDensity: A Hyper-Volume-Based Density Metric for Automatic Cluster Evaluation

Dylan Molinié[0000−0002−6499−3959], Kurosh Madani, and Abdennasser Chebira

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. As the systems produce ever more, becoming more and more complex and universal, the time spent on product validation narrows. The main alternative to the examination of every final product is a statistical study, with a confidence interval; another solution may be an automated validation, possibly integrated to the system's core. Some works addressed this topic using unsupervised learning, and especially clustering, but one challenge remains, that of supporting the information brought by these blind, automatized techniques. In this chapter, we address this problem of cluster validation using a data-driven metric, based on the hyper-volume theory, to estimate the clusters' density, so as to estimate how representative they are. We apply this metric to real industrial data, and show how resilient and representative this metric is, and we extend it to two other metrics from literature to provide a hybrid quantifier, which allows to automatically validate or reject a cluster. The accepted clusters can eventually be assimilated to true regions of the feature space, whose representativeness and meaningfulness are given by this hybrid quantifier.

Keywords: Unsupervised clustering · Knowledge extraction · Automatic characterization · Cluster density · Hyper-volume theory.

1 Introduction

With the ever increasing amount of data generated, collected and finally stored, it becomes less and less feasible to correctly analyze them all. More data means higher processing-time, more energy, and, perhaps, lower interpretable results (but with a higher representiveness) [16]. This is not new though, and many policies have been developed to reduce the processing time: on-the-fly processing [1], dimensionality reduction [8], off-line processing [12], etc.; but the many more data now require innovative ways to help analyze them rapidly and accurately, to make the most of them.

In the past, the only one way to deal with data was by a manual expertize; this is generally more accurate (since an expert asserts every information drawn), but is long, expensive, and less and less feasible due to the increasing amount

of information [3]. To compensate that, a new trend has been gaining ground for the four last decades with the emergence and generalization of the Machine Learning paradigm, and its related fields [4].

The one which interests us the most here is Data Mining, a field of research which consists in digging into data so as to extract some sort of information from them, whether automatically (unsupervised learning) or driven by any sort of expert (supervised learning) [19]. Among a thousand tools, unsupervised clustering is one of the most useful to provide a first insight on the data, by automatically and blindly gathering them into compact and homogeneous groups, within which they share some similarities, but not from a group to another; notice that clustering can also be supervised, but is essentially used to find a fine border between already-known classes, and, to this end, requires much information upstream [11].

Actually, unsupervised clustering can help understand better the system, by discovering non-salient information or patterns, and can also be used to simplify the analysis of its data, by diminishing their number (while keeping their representativeness) or just acting as a pretreatment to rough out the system [13].

Nonetheless, the fact that unsupervised learning needs no reference to operate is both its strongest advantage and its greatest weakness. Indeed, if one already knows the system, there is no need to use such knowledge extraction tools, since that knowledge would already be accessible; such knowledge is often long and hard to build though, whence the recourse to unsupervised learning. However, operating in a blind context makes the validation harder, since there is no way to confirm the validity and representiveness of the extracted knowledge [22].

Focusing on unsupervised clustering only, blindly asserting a cluster is not an easy task, due to the absence of already-defined objectives to which compare the clusters to; however, the clusters must be asserted in some fashion, to estimate their quality (outliers, overlaps, etc.) and representativeness (clear, distinct and distant borders, and meaningful cluster, i.e., a cluster represents something real in the system) [28]. This estimation must be done in some fashion, and since there is no reference to which compare the clusters, it should be done by unsupervised and automatic metrics/quantifiers. This becomes clearly noticeable when dealing with real data, issued from complex and nonlinear systems, such as the industrial systems for instance; as an example, [5,20] used clustering to identify the real behaviors of systems, and asserted the clusters through such metrics.

From literature, some of such quantifiers have been proposed so far to estimate the quality of the clusters, either based on their homogeneity, compactness, isolation, etc. In [21], we proposed a new metric, based on the hyper-volume theory so as to define and compute the "density" of n-dimensional groups of data, which serves as a compactness indicator. In this present chapter, we continue this work, by generalizing this metric and by applying it to real industrial data (related to Industry 4.0) so as to test it in real situations.

This chapter is organized as follows: section 2 introduces the most common quantifiers from literature, section 3 describes our methodology and the proposed density-based metric, section 4 consists of a set of applications to show the benefit of using this metric, and section 5 eventually concludes this chapter.

2 State-of-the-Art

This present work aims to assess the quality of groups of data, and especially that produced by unsupervised clustering. As a consequence, the very first matter to discuss is clustering *sensu stricto*. It is a powerful tool, which optimizes the inner organization of a dataset; its aim is to gather data into compact groups, whose inner-distances (i.e., the distance from a data to another inside the same group) should be as small as possible, whilst the outer-distances (i.e., the distance from a data of a given group to another in a different one) should be as large as possible. Therefore, the main challenge of unsupervised clustering is to automatically find the optimal borders between the groups of data, so as to minimize the sum of all the intra-cluster distances, whilst maximizing the sum of all the inter-cluster distances, both at the same time.

From a mathematical point of view, clustering is a matter of optimization; however, one must keep in mind that clustering should also have a real meaning: the clusters should represent the intrinsic properties of a system. As an example, clustering was used in [23] to identify the real behaviors (and possible failures) of real industrial systems, by gathering data with similar states over time.

As a mathematical optimizer, one may quite confidently trust a clustering algorithm, for it was designed to solve that optimization problem with great intelligence; of course, some are poorly or naively designed, but many carry out their task with brio. The quality and representativeness of the clusters depend on two major elements: the data complexity and the considered feature space.

Regarding the former, it is quite natural: some algorithms are elementary, and just linearly separate the data, which gives good results with simple and linearly separable datasets, but less with more complex and nonlinear ones.

Concerning the feature space, it is a little less evident: the feature space is a very important key parameter of clustering, since two data might be close to some extent in a given feature space, while also being very distant in another. Consider for instance a sheet of paper: if laid flat, two corners are very distant from one another, but if folded up on itself, the two corners are stuck to one another. This simple example illustrates the difference between a Cartesian space (flat sheet) and a folded up space (folded up sheet). The outputted clusters will manifestly not be the same in these two feature spaces.

Even tough that is not a rule by itself, a possible feature space for a real industrial system can be that whose basis is composed by its different sensors; the problem is that such system typically comprises hundreds of sensors, which implies that the corresponding feature space has a very large dimensionality, making greatly harder the interpretation of the clusters. As such, it is unfortunately not always easy to interpret the outputted results, for it is all about the considered feature space; it may be simple to understand why some data have been gathered when only two dimensions are considered, but that becomes more blurry as the dimensionality and the complexity in the data increase.

There exists many clustering algorithms (Neural Gases [18], Region Growing [25], Correlation Clustering [2], etc.), all with their advantages, drawbacks, and dedicated contexts. Since we will deal with real industrial data in this chapter,

we will discuss some previous works conducted on this topic, in order to select an appropriate unsupervised clustering method for our experiments.

In [20], three methods were investigated and compared: the K-Means (KMs) [17], the Kernel K-Means (KKMs) [7] and the Self-Organizing Maps (SOMs) [14]; applied to real industrial data, the experiments showed that the SOMs achieved great accuracy in pointing out the real behaviors of the system, followed by the KKMs, and the KMs eventually provided the poorest results.

Following a similar methodology, [22] introduced the Bi-Level Self-Organizing Maps (BSOMs), a two-level clustering method based on the stratification of several SOMs; compared to the original SOMs and to the K-Means, the BSOMs seemed more resilient and robust in the identification of the system's behaviors. It is this last method that will be used in the rest of this chapter to build the clusters and eventually estimate their quality with the proposed metric.

When only considering unsupervised clustering algorithms, their common problem is the validation of the outputted clusters; indeed, in the absence of reference, how to estimate the quality and representativeness of a cluster? Several works have addressed this question, by proposing data-driven metrics, quantifying how good is a cluster, in the sense of the considered metric (compactness, homogeneity, isolation, etc.). There exists two categories of indicators: that based on the ground truth, called "extrinsic metrics", and that which only use the data, assuming no previous knowledge, called "intrinsic metrics" [31]. In this chapter, we deal only with unsupervised tools, and assume we have access to as little information as possible, and thus we will only consider the second category.

The first metrics for cluster's quality evaluation were statistical: close to the spirit of clustering itself, the idea was to estimate how near the data within a cluster were, and how far from those of a different cluster they were. The first quantifier was proposed in [9] and is known as Dunn Index: its definition can roughly be the ratio between the minimal inter-cluster distance and the maximal inter-cluster distance; this metric is very simple, but not very representative of the clustering, since it only considers extremal values, and, as such, even with a perfect clustering except one outlier (such as a lone cluster overlapping another), the Dunn Index would be very low. To generalize it a little, [6] proposed the Davies-Bouldin Index, which acts very similarly, except that the minimum and maximum are replaced by means; the idea is, somehow, to average the Dunn Index computed for every pair of clusters, which allows to diminish the impact of an outlier. Finally, a third index was developed in [26], namely the Silhouette Coefficients, more sophisticated, more representative, but also very time-greedy, and hardly applicable to large datasets; these coefficients take into account every pair of data to estimate both their intra-cluster and inter-cluster distances and somehow average all of them and eventually propose a normalized value assessing both cluster's compactness and isolation.

For quite a long time, the Silhouette Coefficients served as the reference for cluster's quality estimation, and few additional works have addressed this problem; it is nonetheless worth mentioning [27] which proposed the Average Standard Deviation, a simple metric based on the computation of the standard

deviation within any axis of the feature space, and then a mixture of some kind of all these standard deviations (minimum, maximum, mean, etc.). This metric is not representative of the isolation of the clusters, but indicates their compactness, and is very fast to compute, especially compared to the Silhouettes.

Finally, we proposed a metric based on the density of the clusters, introduced in the original conference paper [21] of which this present chapter consists in a fully revised version; its aims was to provide a definition of the density of n-dimensional groups of data. Several attempts were made to provide such definition, but most stuck to elementary concepts: cluster's span (maximal distance between two points within a cluster), maximal distance to the cluster's barycenter, neighborhood (adjacent data belong to the same cluster, and two clusters are distinct if they have no data distant of a least a certain value), etc. We proposed to stick to the definition of Physics, as the ratio between the number of items (thus data) and the "volume" of the cluster. To define such volume with n-dimensional data, we used the hyper-volume theory, which provides n-dimensional equivalent to the traditional 3D volumes (cube, sphere, etc.). This definition was applied to clusters, and helped characterize them, by attributing them a density score: the higher the denser. This metric proved to be very representative and indicated the very compact clusters, and the "problematic" ones. Nonetheless, it is quite sensitive to outliers, which may produce false positives.

3 Methodology

The core of this chapter is the automatic evaluation of groups of data, in order to provide some criterion indicating their are correctly built, well isolated, compact, dense, etc., and state whether a further refinement is needed or not. If a cluster obtains bad scores, it may be worth being reworked in some fashion (refinement, outliers removal, new clustering, etc.); on the contrary, if the indicators are good, the cluster should be assumed to be representative and meaningful, and, as such, should not be selected for further refinement.

This should be seen as a pretreatment for any algorithm which would rely on clustering as a first step. An example can be found with the Multi-Modeling of systems [29]: it consists in splitting a dynamic system into pieces and then propose local models, eventually connected in some fashion. The system splitting can be manually done, but clustering its feature space can be a good, cheaper and faster alternative: as an example, with a real industrial system, clustering may help point out and isolate its distinct regular states, which can be seen as its "behaviors". As a consequence, the multi-model would be a complex and dynamic combination of the behaviors of the system, which can serve several purposes, among which modeling, monitoring or even prediction.

In this chapter, we will consider three intrinsic metrics introduced in section 2, namely the Average Standard Deviation (AvStd), the Silhouette Coefficients (SCs) and the proposed one, that based on the hyper-volume theory and the physical density, that we will refer to as Hyper-Density (HyDensity). Each method will be explained in detail, and will be used within section 4.

In all the following, we will refer to the database as $\mathcal{D} = \{x_i\}_{i\in[\![1,N]\!]}$ with $N = |\mathcal{D}| = \text{card}(\mathcal{D})$ the size of the database, and n the data dimensionality, to $\mathcal{C} = \{\mathcal{C}_k\}_{k\in[\![1,K]\!]}$ the set of clusters, with K the total number of clusters, and to d as a mathematical distance.

3.1 Already-Existing Metrics

In this subsection, we detail two already existing metrics, namely the Silhouette Coefficients and the Average Standard Deviation, to which the proposed density-based metric will be compared for validation in the next section.

Silhouettes [26]: these coefficients can be seen as an evolution of the Davies-Bouldin Index, which is itself an ameliorated version of the older Dunn Index. The idea is similar for all three: estimate how close the data within a cluster are, and how distant the data from different clusters are. The Dunn Index only considers the extremal values, and is thus lowly representative, since easily confused by outliers; notice there is one unique score for the whole clustering. Similarly, the Davies-Bouldin Index provides a unique score for a full set of clusters, but it considers the mean interactions between each, and, as such, it averages the outliers. Finally, that which interests us the most, the Silhouette Coefficients (or just Silhouettes) are a dynamic comparison between any pair of data; this comparison is exhaustive, accurate, but also long, since the number of pairs is quadratic with respect to the number of data. There is one Silhouette Coefficient per pair of data, and the final score of the clustering is given by a mixture of any kind of all these coefficients. They are obtained in three steps:

1. For any data x_i, compute its average distance $\text{avg}(x_i)$ to any of its neighbors in the same cluster $\mathcal{C}_k$ (1); that measures the compactness of the clusters, i.e., if the data contained within are close from one another: the lower, the more compact, with 0 the best case (data overlay).
$$\forall x_i \in \mathcal{C}_k,\ \text{avg}(x_i) = \frac{1}{|\mathcal{C}_k| - 1} \sum_{x_j \in \mathcal{C}_k \setminus \{x_i\}} \text{d}(x_i, x_j) \tag{1}$$
2. For every different cluster $\mathcal{C}_{k'}, k'\neq k$, compute the mean distance between x_i and any data belonging to $\mathcal{C}_{k'}$: the minimal value is the dissimilarity score $\text{dis}(x_i)$ of x_i (2); that represents the isolation of the clusters, i.e., if the data of a given cluster are far from that of the other clusters: the higher, the more distant and, thus, the better (clear borders), with 0 the worst case.
$$\forall x_i \in \mathcal{C}_k,\ \text{dis}(x_i) = \min_{\substack{k'\in[\![1,K]\!] \\ k'\neq k}} \left\{ \frac{1}{|\mathcal{C}_{k'}|} \sum_{x_j \in \mathcal{C}_{k'}} \text{d}(x_i, x_j) \right\} \tag{2}$$
3. The Silhouette Coefficient $\text{sil}(x_i)$ of x_i is finally obtained by subtracting the average distance $\text{avg}(x_i)$ from the dissimilarity $\text{dis}(x_i)$, and by dividing this value by the maximum among both indexes (3).
$$\forall x_i \in \mathcal{C}_k,\ \text{sil}(x_i) = \frac{\text{dis}(x_i) - \text{avg}(x_i)}{\max\{\text{dis}(x_i), \text{avg}(x_i)\}} \tag{3}$$

A Silhouette measures how correctly a data is classified; it can somehow be interpreted as a regulated dissimilarity measure, reduced by the average distance as a penalty: for a given cluster, this measure is higher if the data contained within are near from each other (avg), whilst being far from that of the other clusters (dis), and vice-versa. The division by the maximum among the two scores has a normalization purpose, so as to bring the Silhouettes in the interval $[-1, +1]$; this eases the interpretation: -1 means x_i is closer to the data of another cluster than its own, and $+1$ means x_i overlays its same cluster's neighbors and is far from the other clusters; in a nutshell, the closer to $+1$, the better.

AvStd [27]: when dealing with statistics, the notion of mathematical moments should spontaneously come in mind, and especially the mean and variance; they are the two most common tools to study a dataset, to inform the user about its inner organization [30]. In particular, the standard deviation informs on data scattering, to know how closely centered around their mean they are. This information is highly valuable, especially when one wants to evaluate the compactness of the group of data, but it unfortunately works only in one dimension. This is not totally true, since a standard deviation relies on a distance, which can easily be n-dimensional, but this indicator is poorly representative when considering several dimensions at the same time, since data may be scattered along an axis, but not along another; the true challenge is to find a way to apply the second mathematical moment to a basis with n axes. A piece of solution was proposed by [27], by computing the standard deviation in every dimension and eventually merging these n values in some fashion rather than directly computing it in that n-dimensional space. This simple change avoids some aberrations, since it is possible to detect that there is a problem along a given axis, but not along another. There are several ways to merge a set of values (the n standard deviations), and three were tested: the minimum, mean and maximum; the author came to the conclusion that their mean is generally the most representative choice, and provides a good (and very fast) estimate of cluster's quality, fairly enough for a quick validation, or with simple databases. This metric was named AvStd after Average Standard Deviation, for it uses the mean of the standard deviations, computed along each dimension. This metric diminishes the impact of the outliers, but is more an indicator of data scattering than anything else. Assuming the center m of a group of data $\mathcal{D}$ is given by (4), the standard deviation $\sigma^{(j)}$ along axis $j \in [\![1, n]\!]$ is defined by (5), and the AvStd measure of the dataset $\mathcal{D}$ is finally given by (6).

$$m = \frac{1}{|\mathcal{D}|} \sum_{x_i \in \mathcal{D}} x_i \tag{4}$$

$$\sigma^{(j)} = \frac{1}{|\mathcal{D}|} \sum_{x_i \in \mathcal{D}} \left(m^{(j)} - x_i^{(j)} \right)^2 \tag{5}$$

$$\text{AvStd} = \overline{\sigma} = \frac{1}{n} \sum_{i=1}^{n} \sigma^{(j)} \tag{6}$$

3.2 Proposed Metric: HyDensity

The notion of density can be found in different contexts: Physics, Chemistry, Human sciences, etc., each coming with its own definition. Nonetheless, it can widely be defined as the ratio between the number of things (mass, particles, people, etc.) and the space these items occupy (surface, volume, land, etc.) [10]. More specifically to Physics, to be a true density, this value must be divided by a reference, so as to normalize it and cancel its units [15,24]; otherwise, one should rather talk about specific mass.

In our case, the specific mass ρ' of a dataset can be given by the ratio between the number N of data instances contained within, and its volume V; the density ρ is that specific mass divided by a reference ρ_{ref}, as expressed by (7).

$$\rho' = \frac{N}{V} \qquad \text{and} \qquad \rho = \frac{\rho'}{\rho_{ref}} \tag{7}$$

The notion of volume is a little ambiguous though, since there is no universal definition for it in computer science. It may be the sum of the distances between any pair of data within, the maximal span of the cluster, or else the shape which covers all these data. Since we dealt with experimental data in [21], we rejected a too strict definition, and preferred to use a more traditional and more universal one. For these reasons, we decided to represent datasets by regular shapes, such as cubes or spheres, and then assimilate the cluster's volume to that of its surrounding and representative shape.

This works fine with 3 dimensions, since there is no ambiguity to define a volume there; but that becomes less common as dimensionality increases. A piece of solution may come along the hyper-volume theory, which provides n-dimension equivalents to the regular 3D volumes (tetrahedron, cube, prism, pyramid, sphere, etc.); for each, a n-dimension equivalent exists, even though it may be hard to mathematically formalize them all. The remaining challenge is to select that which represents best the data.

Nonetheless, two shapes stand out: the cube and sphere, for they are natural. Indeed, if one wants to fulfill a Cartesian space, leaving no uncovered areas, cubes should be considered; this is for instance the reason why the moving boxes are squared. Nevertheless, Nature does not really agree: its elementary shape is the sphere, as can be seen with the celestial bodies and atoms; it is very relevant due to its democratic aspect: any point on its surface is at the exact same distance from its center. Actually, if there is no particular reason to give a certain shape to something, a sphere-like one is probably the most universal choice.

In our previous work, we searched for the smallest hypersphere containing the whole dataset, and then assimilate their respective volumes. The volume of the n-hypersphere of radius r is given by (8), where Γ is the Gamma function.

$$V^{(n)}\big(r\big) = r^n V^{(n)}\big(r=1\big) = \frac{r^n.\pi^{n/2}}{\Gamma\big(\frac{n}{2}+1\big)} \tag{8}$$

Unfortunately, a group of data is rarely perfectly regular, thus it remains to evaluate the radius r of that hypersphere; the most natural value may be the

maximal distance separating two of its points, which can be seen as the cluster's span, but this is heavy to compute (complexity of $\mathcal{O}(N^2)$). Instead, we proposed to use an intermediate (but representative nonetheless) value for the radius: we preferred the maximal distance between any point and a reference (we chose the dataset's barycenter); this allows to reduce the complexity to $\mathcal{O}(N)$. The cluster's span corresponds to the true smallest hypersphere containing the cluster, whilst the maximal radius provides an upper estimate of the cluster's volume, but is easier and faster to compute. Moreover, a cluster's barycenter is essentially a region of influence, which only depends on the recorded data: tomorrow, a data belonging to a given class could slightly differ from its historical neighbors; therefore, an estimate by excess is not a so strong assumption in that case. The radius r of cluster $\mathcal{C}_k \subset \mathcal{D}$, whose mean is m, is given by (9); the span s can be estimated as the double of that radius r.

$$\forall \mathcal{C}_k \subset \mathcal{D},\ r^{(mean)} = \max_{x_i \in \mathcal{C}_k} \left\{ \mathrm{d}(x_i, m) \right\} \tag{9}$$

Notice that this estimate is sensitive to outliers, since the radius is defined by most distant point from the center; to decrease the impact they may have, one may identify several outermost points, and then compute their average.

The density metric estimates the compactness of a cluster, and thus its homogeneity: the higher, the better. It may take any value between 0 and $+\infty$, thus it may be difficult to interpret the results. As as consequence, and similarly to the Silhouette Coefficients, the idea is to normalize that value by a reference, which would bring back the density measure into the standard interval of $[0, 1]$, with 0 the worst case, and 1 the best case. A possible candidate may be the dataset's density itself, but a more judicious reference is the maximal density found within the clusters, ensuring that the maximal normalized value is 1.

Indeed, since this metric serves to characterize clusters, one may assume that at least one of them is well-built and compact. This cluster can thus serve as a reference, and its own density can be used as a candidate for ρ_{ref} in (7).

As a consequence, we finally define HyDensity (standing for Hyper-Volume-based Density, or just Hyper-Density) as the normalized specific mass ρ'_k of cluster $\mathcal{C}_k$, i.e., divided by the maximal of these specific masses among of the clusters, as given by (10).

$$\mathrm{HyDensity}(\mathcal{C}_k) = \rho_k = \frac{\rho'_k}{\max\limits_{i \in [\![1,K]\!]} \{\rho'_i\}} \tag{10}$$

Notice that, for comparison, we will also consider the hyper-sphere defined by the real cluster's span, i.e., the maximal distance between any pair of its points, as defined by (11).

$$\forall \mathcal{C}_k \subset \mathcal{D},\ r^{(span)} = \frac{1}{2} \max_{x_i, x_j \in \mathcal{C}_k} \{\mathrm{d}(x_i, x_j)\} \tag{11}$$

Therefore, we define $\rho^{(mean)}$ the Hyper-Density based on $r^{(mean)}$, and $\rho^{(span)}$ that which uses $r^{(span)}$ instead.

3.3 Hybridization of the Metrics

Following our works in [21] and [23], and with respect to all what have been said concerning the three metrics discussed above (SCs, AvStd and HyDensity), we propose a hybridized version of them, which uses the strength of each and overcomes their respective weak spots.

Ironically, what a metric sees is often what another misses, and vice-versa; nevertheless, nothing prevents us from merging them within a decision tree. HyDensity informs on the density of a cluster, i.e., if its data are numerous and close from one another: if yes, it is probably compact and of good quality, but, on the contrary, a low score means the presence of outliers or of scattered data. AvStd indicates if the data are centered around their mean or if they spread over a large area; combined to HyDensity, it tells if a problematic cluster is only due to the presence of some outliers, or if it contains few, highly scattered data. Additionally, the Silhouettes inform on the cluster's isolation and homogeneity.

Both HyDensity and AvStd inform on the intrinsic quality of the dataset, i.e., if their data are compact and homogeneous, whilst the Silhouettes indicate if it is clearly distinguishable (separated) from the others. If a cluster is of good quality and distant from the others, a real group of data, with an intrinsic meaning (i.e., a true class), is detected. On the contrary, depending on the scores of the metrics, the cluster may be worth being reworked or just left alone for further usage. This hierarchy is summarized as a decision tree and presented as of Fig. 1.

To keep it clear and understandable, we will refer to this metric as HyDAS, for Hybridized Density-AvStd-Silhouettes-based metric.

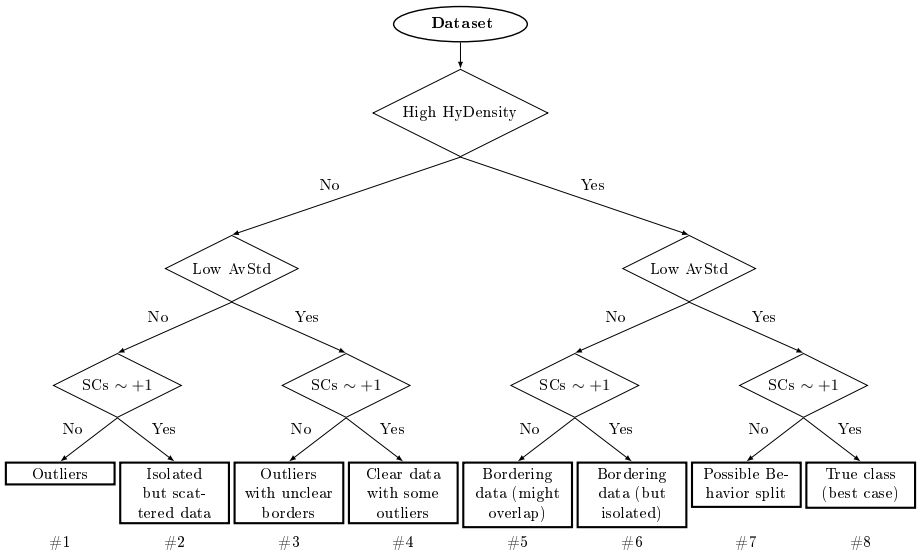

Fig. 1. The HyDAS metric, a hierarchical hybridization of the three considered metrics. The number #X below every leaf is the case number (for indexation).

4 Application to Industrial data

In the original work [21], the cluster's density was assessed on academic datasets. These datasets were clustered using Self-Organizing Maps, and the proposed metric proved to be meaningful, accurate and helped identify both correct and problematic clusters. In this present chapter, we extend the experiments to real industrial data, we test the two radius estimations (max-to-mean and max-span) to compare their respective accuracy, we confront HyDensity with both AvStd and Silhouettes, and we assess the validity of the hybridized metric HyDAS.

Notice that the scoring of AvStd is the reverse of that of HyDensity and Silhouettes: for the latter, the higher the better, whilst it is the other way around for the former. In order to standardize the interpretation, we will consider the reverse of AvStd instead, so as to have the three metrics behave the same way: the higher the better. Moreover, similarly to HyDensity and to Silhouettes, having a normalized value is easier to interpret; for that reason, we divide by the maximal reversed value among all the clusters to obtain the final, normalized measure. By denoting the original AvStd measure of cluster $\mathcal{C}_k$ by $\overline{\sigma}'_k$, the corrected measure, that we use in all the following, is given by (12).

$$\overline{\sigma}_k^{-1} = \frac{1/\overline{\sigma}'_k}{\max\limits_{i \in [\![1,K]\!]} \{1/\overline{\sigma}'_i\}} \tag{12}$$

In all the following, we will refer to the identified clusters by a unique color; this color scheme is given as of Table 1. For instance, saying "clt1" or "blue cluster" refers to the exact same cluster in both cases.

Table 1. Clusters' color scheme.

Clusters	clt1	clt2	clt3	clt4	clt5	clt6	clt7	clt8	clt9
Colors	Blue	Orange	Green	Red	Purple	Pink	Gray	Olive	Cyan

4.1 Academic Dataset: Gaussian Distributions

Since the methodology is not trivial, it is worth showing how it works through a simple and didactic example; for this reason, we first experiment a set of low-dimensional Gaussian distributions. These small datasets represent the most common scenarios one may encounter when dealing with groups of data: isolation, closeness, overlaps and stretching. Since clustering is operated in the feature space, it is useful to know how the data are related to each other, dimension by dimension; indeed, two data may be close in one of them, but not in another.

The database is a set of twelve Gaussian distributions, along three dimensions (denoted as x, y and z); each set comprises 250 data and, for each dimension, is randomly attributed a mean between 0 and 1 and a standard deviation of 0.05.

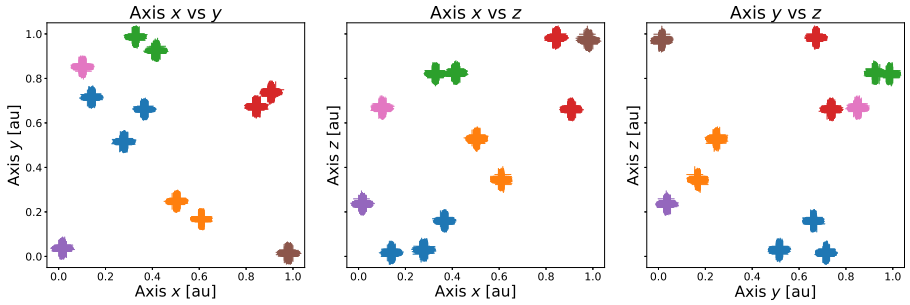

Fig. 2. The twelve Gaussian distributions, clustered by a 3×3 BSOM.

This database has been clustered using a BSOM, with 10 SOMs of size 3×3, resulting in seven nonempty clusters; the corresponding clustered database is depicted on Fig. 2, where the data are projected onto the three plans formed by any pair of the basis' axes, and where every color represents a unique cluster (consistent within the graphs). Notice that "[au]" stands for "Arbitrary Unit", for the intrinsic values and physical dimension of the axes do not matter here.

On that figure, four types of clusters appear: 1) The quite close sets, but scattered though, not completely isolated from the other groups (blue); 2) The sets close in only one or two dimensions (orange and red); 3) The gathering of several sets, close and isolated in any dimension (green); 4) The lone sets, compact, dense and isolated (purple, brown and pink).

For all these clusters, their corresponding quantifiers are regrouped within Table 2. As a remainder, $\overline{\sigma}^{-1}$ is the normalized reversed AvStd, SCs are the mean Silhouette Coefficients, $\rho^{(mean)}$ is HyDensity using the mean-to-max distance as hyper-sphere's radius and $\rho^{(span)}$ is that using the maximal cluster's span, and HyDAS is the hybridized metric. All the quantifiers are normalized between 0 (worst) and 1 (best), with no physical dimension, and the HyDASes are a score from 1 to 8, corresponding to the different cases explained on Fig. 1. The results computed over the database have been added for comparison, with its SCs set to 0 by default (as there is one unique dataset, there is no dissimilarity measure, thus the Silhouettes can not by applied to a lone dataset).

Beginning with the database, its quantifiers (with the exception of the SCs, which can not be computed there) achieved poor results, both regarding its HyDensty and its AvStd: the data are scattered and lowly compact; this is not surprising though, with respect to the original database: there are many empty spaces, and the Gaussian sets are locally distributed, *ipso facto* leading to scattered (low $\overline{\sigma}^{-1}$) and stretched (low $\rho^{(mean/span)}$) data. It is with no surprise that the HyDAS scores are 1 in each case, indicating non-homogeneous data.

Regarding the clusters issued from the BSOM, there is much to say. First, as observed in the introduction of the database and of the clusters, that which comprise several local Gaussian sets, and that we consider as non-compact, got poor results. For instance, clt1 comprises three quite close, but not overlapping

Table 2. Normalized metrics of the Gaussian clusters.

Clusters	$\rho^{(mean)}$	$\rho^{(span)}$	$\overline{\sigma}^{-1}$	SCs	HyDAS (mean)	HyDAS (span)
dba	0.44	0.47	0.02	0.00	1	1
clt1	0.67	0.68	0.07	0.74	1	1
clt2	0.69	0.68	0.09	0.82	2	2
clt3	0.81	0.80	0.20	0.83	6	6
clt4	0.63	0.62	0.07	0.72	1	1
clt5	0.97	0.99	1.00	1.00	8	8
clt6	0.94	0.96	0.92	1.00	8	8
clt7	1.00	1.00	0.91	0.98	8	8

though, sets of data: there is an empty space between them, in every of the three dimensions. As a consequence, it is with no surprise that the three quantifiers got intermediate values, for it depends on the tolerance one accepts: the three sets are quite close, thus they may be considered as representatives of the same class; but they are scattered nonetheless, thus if the tolerance is low, this previous assertion should be rejected, and the cluster should be reworked. This is exactly that the HyDensities (mean and span alike) and the SCs tell us; notice that the corrected AvStd is very bad. Actually, it is the HyDensities and Silhouettes which are right there: the cluster is not perfect, but it may represent a unique class nonetheless, if the tolerance on the classification is reduced a little.

These observations work equally for clt2 and clt4, which share the same state than clt1: two Gaussian sets, not so distant but not touching tough, and quite correctly isolated from the others, thus they may be assimilated to unique classes if the tolerance is, anew, reduced, but if one wants a very accurate clustering, they should be reworked, as indicated by HyDAS. On the contrary, clt3 is a good example where HyDensity and the SCs overcame AvStd: indeed, the two former ones got very good results, but not the latter, although the green cluster comprises two very close sets, even overlapping in two of the three dimensions. This cluster is clearly isolated from the others, and is compact and dense, thus this assimilation is very likely a good thing there, even though AvStd tells not, which is probably a mistake here. Anyway, HyDAS gives a good score nonetheless, indicating the cluster may be worth being left alone for possible future usage.

Finally, the last clusters, i.e., the three correctly identified (one cluster for one set), namely clt5, clt6 and clt7, got the highest results, very close to 1 for every quantifier, which is actually true, since there is only one cluster for one unique set (thus class). For the three, HyDAS got the highest score, indicating that the clusters are of very good quality, compact, dense, homogeneous and isolated from the others, and should *ipso facto* be assimilated to true classes, which is actually true.

Notice that the two first columns of Table 2, i.e., the two versions of the HyDensity measure, got very close results; the max-to-mean version generally achieved slightly lower values, due to the estimation by excess of the hypersphere's volume, leading to a slightly lower density compared to a finer radius

obtained with the maximal span. Here, in every case, both versions stated the exact same thing on every cluster, but the max-to-mean version is easier and faster to compute, thus this emphasizes that this version is a good candidate for the computation of the Hyper-Density of the clusters.

Finally, which are the conclusions drawable from this academic example? First, the metrics globally helped identify both correct and problematic datasets; when one cluster contained compact and homogeneous data, the three quantifiers got very high values (close to 1), indicating a correct clustering, and the HyDAS score confirmed that as well; on the contrary, when the data were scattered and/or stretched, the quantifiers got lower values, indicating non-compact groups of data, which is actually true, confirmed by the HyDAS scores (0 or 1). Moreover, AvStd mistook only once, whilst both HyDensity and the Silhouettes were correct with the seven clusters. The main conclusion may be that AvStd is a good estimation for validation, but the most trustful metrics are undoubtedly HyDensity and Silhouettes, but the first is greatly faster to compute, and may *ipso facto* be the first candidate for cluster characterization.

4.2 Real Industrial Dataset

Now that the metrics are assessed over an academic example, it is time to test them in real situation; to that purpose, we apply them to the automatic characterization of a blind clustering operated over real industrial data.

This work takes place in the European project HyperCOG, whose aim is to study the cognitive plant, with an Industry 4.0 vision. This project acts on two levers: a cyber-physical system which should link all the production units and control systems, so as to have an oversight on the whole plant, and control almost anything from anywhere; and highly intelligent tools and algorithms, able to analyze and interpret the evolution of the different systems and react in case of need. This book chapter focuses on the second part of the project, i.e., the cognitive and intelligent framework oriented toward the cognitive plant.

To test the works, several industrialists provided us with their factory's data; one of them is Solvay®, a chemical company developing and producing many specialty products. The database they provided comprises two hundreds of sensors, containing 105,120 samples each (recorded one per minute). Dealing with blind data is often tricky, thus we asked them to diminish the number of sensors, by pointing out the most representative ones, reducing the database to fifteen sensors of major importance (situated at key stages in the production chain); we will only consider these sensors to ease the interpretation. Five of them are depicted on Fig. 3; the ten others are not represented for the sake of clarity. Each sensor is compared to the others to show their interactions.

On the sub-figures, some groups of data appear: they are colored circled to stand them out. With respect to the work presented in [23], these different groups correspond to the real behaviors of the system, in the sense of its regular and expected states. In this chapter, we do not consider them this way, and we just want to know if these groups can be identified by unsupervised clustering, and validated using the quantifiers introduced in section 3. The purpose is not

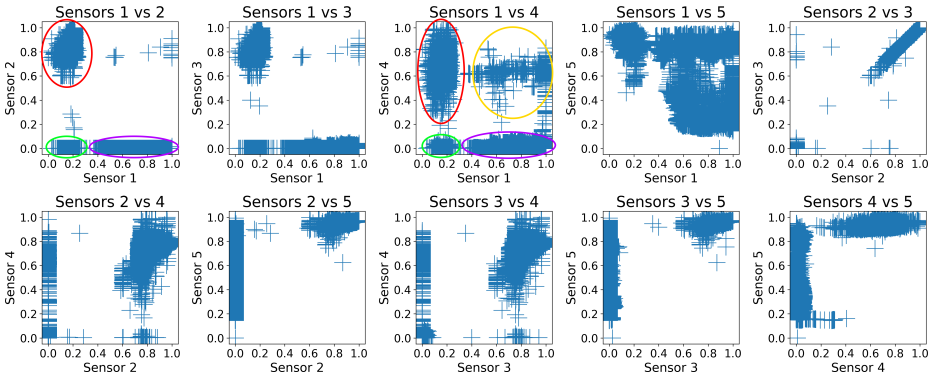

Fig. 3. Five sensors of the Solvay's database and their feature space's compact regions.

so much isolating the clusters as estimating their intrinsic quality, in order to blindly state on their relevance and representativeness.

To that purpose, the database is clustered using a BSOM (comprising anew ten 3×3 SOMs), resulting in four nonempty clusters, all depicted on Fig. 4. Globally, the expected groups of data are identified as unique clusters, even though overlaps exist. For instance, the blue cluster mostly corresponds to the red circle on Fig. 3, but it also comprises the data of the yellow circle; that being said, this may have an intrinsic meaning by itself, which is actually true with respect to the closeness of the blue cluster in most of the dimensions.

The quantifiers of both database and clusters are regrouped within Table 3; anew, the database's Silhouettes are set to 0 as default values. The first thing to notice is its very low AvStd, whilst its densities (both versions) are quite high. Actually, the former result can be explained by the data scattering across all dimensions, as evidenced by the gaps between the Fig. 3's colored circles. The latter can be explained by the high number of data contained in a fixed hypervolume; indeed, the data are repeating themselves through time, thus there are more points, although the feature space remains the same. This last remark may be a great drawback of HyDensity, since redundant samples might confuse it (even though that is also true for the two other metrics).

Concerning the clusters *sensu stricto*, the first thing to notice is the values achieved by clt2, which obtained the best results with every metric, indicating it is very dense and compact; its HyDAS score is 8, meaning it is likely representative of a real class. According to Fig. 4, this cluster mostly corresponds to the purple circle (at least a part of it). The classification as a unique class may be both right and wrong, depending on the tolerance one has on the separation of the system's behaviors. Here, the green and orange clusters actually correspond to two steps of a unique behavior: its initialization and its steady-state, respectively, thus if one wants a fine grain partitioning, having two clusters is acceptable, but maybe not with a coarser grain. In any case, the green cluster, clt3, also achieved honorable results, also obtaining a HyDAS score of 8, as clt2,

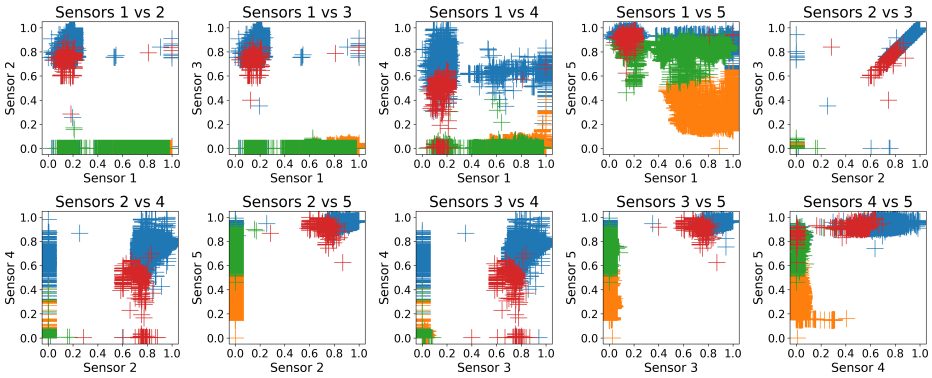

Fig. 4. The Solvay's clustered database using a 3×3 BSOM.

indicating another unique class (actually the other part of the clt2's). Their inner quality is too high to not assimilate them as real classes on their own.

On the contrary, the two other clusters, clt1 and clt4, achieved poor results, with low HyDAS scores (1 and 2, respectively). Indeed, the blue cluster contains data close in some dimensions ("Sensors 1 vs 2" for instance), but not in others ("Sensors 2 vs 4" for instance), resulting in scattered data according to AvStd, whence its very low score. Moreover, due to the high proximity (almost overlaps) of the blue and red clusters in most dimensions, its Silhouette Coefficients are very low, and even negative, indicating a poor clustering; that being said, these bad results are not so much due to the proximity with another cluster as due to a real presence of several sub-classes within, fact that is pointed out by HyDensity, which achieved intermediate results, not so low than AvStd and the Silhouettes. In this present case, it is more a question of coarse clustering rather than true overlapping or mis-classification, fact that is detected by HyDensity only.

It is about the reverse that happens with clt4, for which HyDensity is low, but not AvStd and SCs (the last ones are even good). Actually, the cluster is quite meaningful, and representative to a real behavior of the system, but the low HyDensity score is due to the presence of outliers (borderline data), especially visible on "Sensors 2 vs 4" and "Sensors 3 vs 4", at values around 0.8 in abscissa, and 0 in ordinate for both. The small groups are distant from the cluster's core,

Table 3. Normalized metrics of the Solvay's clusters.

Clusters	$\rho^{(mean)}$	$\rho^{(span)}$	$\overline{\sigma}^{-1}$	SCs	HyDAS (mean)	HyDAS (span)
dba	0.72	0.70	0.15	0.00	1	1
clt1	0.59	0.69	0.23	-0.21	1	1
clt2	1.00	1.00	1.00	1.00	8	8
clt3	0.87	0.84	0.86	0.87	8	8
clt4	0.42	0.55	0.62	0.80	2	2

greatly increasing the cluster's maximal intra-distance (span), decreasing *ipso facto* its Hyper-Density, whence the low values obtained. These small outliers may be worth being isolated, or, perhaps, purely removed, thing that SCs missed, but not AvStd nor HyDensity (and especially the max-to-mean variant).

Therefore, what can we draw from that experiment on real industrial data? First, the Silhouette Coefficients are very meaningful, but may sometimes be mistaken, cases when the two other metrics often indicated something else, and especially HyDensity. This last metric, the proposed one, proved to be meaningful in all cases, and was never wrong. Moreover, the HyDAS scoring proved also to be resilient, and attributed low scores to issuing clusters, either due to the presence of outliers or due to another, close cluster.

5 Conclusion

In this chapter, we proposed Hyper-Density (HyDensity), a blind quantification metric for cluster validation, based on the Physics' definition of the relative density, i.e., the ratio of the specific mass of a body to its volume. As we operated in multi-dimensional spaces, we proposed to use the theory of the hyper-volumes, which defines multi-dimensional equivalents to the more traditional 3D volumes (cube, sphere, etc.); the proposed metric, is therefore defined as the ratio of the number of data instances and their related volume, defined as the smallest hyper-sphere containing them all.

To that end, we first partitioned the database using unsupervised clustering; we chose BSOM, a clustering technique consisting in a projection of several Self-Organizing Maps onto a unique, final one. This method proved to be resilient in the identification of the behaviors of industrial systems.

We confronted HyDensity to two other metrics from literature, namely the Average Standard Deviation[1] and the Silhouette Coefficients. Additionally, we also proposed the Hybridized Density-AvStd-Silhouettes-based metric (HyDAS), a hybrid quantifier based on the three previous ones, specifically crafted so as to give a score to a group of data, and represent its quality and meaningfulness.

We first applied these metrics to an academic example consisting of several 3D Gaussian distributions to test how representative and trustful they are. Applied to the characterization of the clusters outputted by a BSOM, they proved to be meaningful and highlighted both correct and problematic clusters, even though AvStd was sometimes wrong, whilst the Silhouettes and HyDensity were rarely.

Once these quantifiers, and especially HyDensity, tested over this intuitive dataset to explain the results in detail, we applied them to real industrial data, provided by a partner involved in the HyperCOG project. After having clustered the database with a BSOM, the quantifiers were computed to automatically characterize the outputted results. They helped identify two problematic clusters (that which may be worth being reworked or split further) and two correct ones (real system's behaviors, thus the corresponding clusters should be left alone).

[1] Precisely, we used our revised, reversed version of it to standardize its interpretation.

Actually, we saw that the results' meaningfulness is more a matter of user's tolerance regarding the grain of the partitioning. HyDensity was always correct, but both AvStd and the Silhouettes confused at least once. That being said, this is not to say that this metric is a thousand times better than any other, but this strengthens the feeling that it is very representative and trustful in the characterization and identification of both correct and problematic clusters.

We also tested two ways to evaluate the smallest hyper-sphere containing all data: half the maximal distance between two of them, and the maximal distance between their mean and any of them. The former is slightly finer, but longer to compute; the results of both the academic and real databases showed that both versions led to close results in the computation of HyDensity, thus the mean-to-max version may be a perfectly acceptable candidate so as to speed up the algorithms, at the cost of a small sacrifice in accuracy.

Finally, the HyDAS metric showed to be a good indicator of cluster's quality, and of what to do according to its score, as detailed in the tree structure of Fig. 1. According to the studies, HyDensity appears to be the most trustful, then come the Silhouettes, and eventually AvStd, thus so should be the HyDAS's hierarchy, to organize the scores in ascending order of cluster's quality. In current version, AvStd is at the second place, thus more weight is put on it, which may confuse a little the scoring. That was made on purpose, since the Silhouettes are generally long to compute (or just impossible, as with a lone database for instance); putting them in the leaves of the hierarchy allows to remove them with ease if required, without changing the tree, so as to work only with HyDensity and AvStd.

As a summary, we introduced HyDensity, a density-based metric for automatic characterization of clusters, which proved to be accurate and resilient in most of the tested cases. Compared to AvStd and the Silhouettes, it showed to be superior to the former, and equivalent to the latter (or even superior), but has the great advantage of being greatly faster to compute. We also presented HyDAS, a hybrid score evaluating the state of the cluster and what to do with it to improve it, by just reading and comparing the three considered metrics.

One very interesting purpose of this metric may be the hierarchical characterization of clusters, by indicating the issuing ones, i.e., that which may be worth being reworked further, so as to obtain a finer clustering. It may therefore be used in a split-and-merge fashion so as to find back the real objective clusters, which can be very helpful in blind and unsupervised contexts in which very few things about the system are known.

Acknowledgements This paper received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 869886 (project HyperCOG-869886). Authors would thank Solvay and especially Mr. Marc Legros for their fruitful discussions. Authors also wish to express their gratitude to the EU 2020 program for supporting the presented works.

References

1. Bavay, M., Fierz, C., Nitu, R.: Data access made easy: flexible, on the fly data standardization and processing. In: EGU General Assembly 2022. Vienna, Austria (May 2022). https://doi.org/10.5194/egusphere-egu22-8262, eGU22-8262
2. Becker, H.: A survey of correlation clustering. Advanced Topics in Computational Learning Theory pp. 1–10 (2005)
3. Ben-David, A., Frank, E.: Accuracy of machine learning models versus "hand crafted" expert systems – a credit scoring case study. Expert Systems with Applications **36**(3, Part 1), 5264–5271 (2009). https://doi.org/10.1016/j.eswa.2008.06.071
4. Buchanan, B.: Can machine learning offer anything to expert systems? Machine Learning **4**, 251–254 (12 1989). https://doi.org/10.1007/BF00130712
5. Calvo-Bascones, P., Sanz-Bobi, M.A., Álvarez Tejedo, T.: Method for condition characterization of industrial components by dynamic discovering of their pattern behaviour. In: ESREL2020 (November 2020)
6. Davies, D., Bouldin, D.: A cluster separation measure. Pattern Analysis and Machine Intelligence, IEEE Transactions on **PAMI-1**, 224–227 (May 1979). https://doi.org/10.1109/TPAMI.1979.4766909
7. Dhillon, I., Guan, Y., Kulis, B.: Kernel k-means, spectral clustering and normalized cuts. KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining **551-556** (July 2004). https://doi.org/10.1145/1014052.1014118
8. Dong, H., Chen, X., Dusmanu, M., Larsson, V., Pollefeys, M., Stachniss, C.: Learning-based dimensionality reduction for computing compact and effective local feature descriptors (2022). https://doi.org/10.48550/ARXIV.2209.13586
9. Dunn, J.: Well-separated clusters and optimal fuzzy partitions. Cybernetics and Systems **4**, 95–104 (September 1974). https://doi.org/10.1080/01969727408546059
10. Encyclopedia Britannica: Density, https://www.britannica.com/science/density, online, last update: February 02 2021. Accessed: October 04, 2022
11. Ezugwu, A.E., Ikotun, A.M., Oyelade, O.O., Abualigah, L., Agushaka, J.O., Eke, C.I., Akinyelu, A.A.: A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Engineering Applications of Artificial Intelligence **110**, 104743 (2022). https://doi.org/10.1016/j.engappai.2022.104743
12. Gan, Y., Dai, X., Li, D.: Off-line programming techniques for multirobot cooperation system. International Journal of Advanced Robotic Systems **10**(7), 282 (2013). https://doi.org/10.5772/56506
13. Golalipour, K., Akbari, E., Hamidi, S.S., Lee, M., Enayatifar, R.: From clustering to clustering ensemble selection: A review. Engineering Applications of Artificial Intelligence **104**, 104388 (2021). https://doi.org/10.1016/j.engappai.2021.104388
14. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics **43**(1), 59–69 (1982). https://doi.org/10.1007/BF00337288
15. Kriegel, H.P., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. Wiley interdisciplinary reviews: data mining and knowledge discovery **1**(3), 231–240 (2011)
16. Lezoche, M.: Formalisation models and knowledge extraction: Application to heterogeneous data sources in the context of the Industry of the Future. Habilitation à diriger des recherches, Université de Lorraine (January 2021), https://hal.univ-lorraine.fr/tel-03178698

17. Lloyd, S.P.: Least squares quantization in pcm. IEEE Trans. Inf. Theory **28**, 129–136 (1982)
18. Martinetz, T., Schulten, K.: A "neural-gas" network learns topologies. Artificial neural networks **1**, 397–402 (January 1991)
19. Mikut, R., Reischl, M.: Data mining tools. Wiley interdisciplinary reviews: data mining and knowledge discovery **1**(5), 431–443 (2011)
20. 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
21. Molinié, D., Madani, K.: 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). https://doi.org/10.5220/0010657500003062
22. 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
23. 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
24. National Aeronautics and Space Administration (NASA): Gas density, https://www.grc.nasa.gov/WWW/BGH/fluden.html, online, last update: May 07 2021. Accessed: October 04, 2022
25. Rabbani, T., Heuvel, F., Vosselman, G.: Segmentation of point clouds using smoothness constraint. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences **36** (January 2006)
26. Rousseeuw, P.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (November 1987). https://doi.org/10.1016/0377-0427(87)90125-7
27. Rybnik, M.: Contribution to the modelling and the exploitation of hybrid multiple neural networks systems: application to intelligent processing of information. Ph.D. thesis, University Paris-Est XII, France (December 2004)
28. Shivakumar, A., Alfstad, T., Niet, T.: A clustering approach to improve spatial representation in water-energy-food models. Environmental Research Letters **16**(11), 114027 (2021)
29. Thiaw, L.: Identification of non linear dynamical system by neural networks and multiple models. Ph.D. thesis, University Paris-Est XII, France (2008), (in French)
30. Wan, X., Wang, W., Liu, J., Tong, T.: Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range. BMC medical research methodology **14**(1), 1–13 (2014). https://doi.org/10.1186/1471-2288-14-135
31. Wang, S., Yu, L., Li, C., Fu, C.W., Heng, P.A.: Learning from extrinsic and intrinsic supervisions for domain generalization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 159–176. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58545-7