The version of record is available online at: http://dx.doi.org/10.1007/s00521-021-05873-3

S.I. : WSOM 2019

Automatic identification of the number of clusters in hierarchical clustering

Ashutosh Karna[1] · Karina Gibert[2]

Received: 24 July 2020 / Accepted: 22 February 2021 / Published online: 13 March 2021

Abstract
Hierarchical clustering is one of the most suitable tools to discover the underlying true structure of a dataset in the case of unsupervised learning where the ground truth is unknown and classical machine learning classifiers are not suitable. In many real applications, it provides a perspective on inner data structure and is preferred to partitional methods. However, determining the resulting number of clusters in hierarchical clustering requires human expertise to deduce this from the dendrogram and this represents a major challenge in making a fully automatic system such as the ones required for decision support in Industry 4.0. This research proposes a general criterion to perform the cut of a dendrogram automatically, by comparing six original criteria based on the *Calinski-Harabasz* index. The performance of each criterion on 95 real-life dendrograms of different topologies is evaluated against the number of classes proposed by the experts and a winner criterion is determined. This research is framed in a bigger project to build an Intelligent Decision Support system to assess the performance of 3D printers based on sensor data in real-time, although the proposed criteria can be used in other real applications of hierarchical clustering.The methodology is applied to a real-life dataset from the 3D printers and the huge reduction in CPU time is also shown by comparing the CPU time before and after this modification of the entire clustering method. It also reduces the dependability on human-expert to provide the number of clusters by inspecting the dendrogram. Further, such a process allows applying hierarchical clustering in an automatic mode in real-life industrial applications and allows the continuous monitoring of real 3D printers in production, and helps in building an Intelligent Decision Support System to detect operational modes, anomalies, and other behavioral patterns.

Keywords Hierarchical clustering · Calinski-Harabasz index · Scalability · Data Science · 3D Printing · Decision Support

1 Introduction

Cluster Analysis is perhaps one of the most researched topics in data mining and unsupervised machine learning. From domains ranging from text, multimedia applications, natural language processing, computer vision, protein-and-genomics data, biological data, social media analytics, and more, clustering is used everywhere in different forms and contexts for it is one of the best techniques to learn typologies in a domain which is crucial in decision support. The core idea of clustering lies in summarizing or segmenting a domain, based on distances between objects or points (and from a wider perspective using dissimilarities). In unsupervised learning scenarios (where no ground truth is available and the main goal is to understand the underlying structure of a domain), clustering is often used as a first step to discover this structure. *Hierarchical clustering* [18] is a classical scheme where underlying relationship among objects is represented hierarchically through a *dendrogram*. The dendrogram shows the inner structure of inertias through clusters at different levels of heterogeneity and provides perspective to understand how many classes

✉ Ashutosh Karna
ashutosh.karna@upc.edu

Karina Gibert
karina.gibert@upc.edu

[1] 3D Printing & Digital Manufacturing, HP Inc., and Intelligent Data Science and Artificial Intelligence Research Centre, Universitat Politècnica de Catalunya-BarcelonaTech, Catalonia, Spain

[2] Knowledge Engineering and Machine Learning Group at Intelligent Data Science and Artificial Intelligence Research Centre, Universitat Politècnica de Catalunya-BarcelonaTech, Catalonia, Spain

the domain contain, at different levels of granularity. Unlike other clustering methods like k-means [17], hierarchical clustering does not need the user to define the expected number of clusters apriori. On the contrary, the number of clusters emerges as a consequence of the clustering process itself and often is determined by visual inspection of the resulting dendrogram (a common praxis among specialists which is in fact related with the optimization of some goodness of cluster indicator, like *Calinski-Harabasz* index).

The hierarchical clustering provides rich information about the structure of the domain and is also robust against outliers and thus comes out to be a preferred choice in understanding the underlying structure of a real domain.

One of the direct application of such an approach could be found in Industry 4.0 [29] applications that revolve around modern data-intensive technologies such as Industrial Internet of Things, Digital-Twins, Virtual-Reality, Additive Manufacturing or 3D Printing, etc, and in particular in contexts where sensor data can be used to identify operational modes, as is the case of 3D printing manufacturing. Indeed, Industry 4.0 proposes complete automation of traditional manufacturing practices largely driven by data and modern technologies. The design principles of Industry 4.0 are discussed in [15]:

- **Interconnection**: The ability of machines, devices, sensors, and people to connect and communicate with each other via the Internet of things.
- **Information transparency**: Access to comprehensive information to inform decisions.
- **Inter-connectivity**: Collect immense amounts of data and information from all points in the manufacturing process, identify key areas that can benefit from the improvement to increase functionality.
- **Technical assistance**: Modern technology to assist humans in decision-making and problem-solving, and the ability to help humans with difficult or unsafe tasks.
- **Decentralized decisions**: The ability of cyber-physical systems to make decisions on their own and to perform their tasks as autonomously as possible. Only in the case of exceptions, interference, or conflicting goals, are tasks delegated to a higher level

Thus, collecting a huge amount of data continuously throughout the smart factory setting and aiding in an autonomous decision are key design principles in Industry 4.0 and require unsupervised learning models to make inferences about the underlying manufacturing processes (like operational modes of any kind of machine such as gas turbines or a 3D printer in digital manufacturing). Using a clustering algorithm for a deep data-driven understanding of the operational modes of a certain machine and later developing a process to recognize the discovered patterns in real production helps in triggering actions automatically. For the reasons mentioned earlier, hierarchical clustering is the most appropriate method to characterize various scenarios in an Industry 4.0 dataset. However, as said before, using hierarchical clustering opens the door to discover any number of existing clusters, provided that an interaction with the specialist to evaluate the resulting dendrogram and determine the number of clusters occurs. Finding the best place to cut the dendrogram and determining the number of clusters, therefore, restricts end-to-end automation, which is fundamental in Industry 4.0. When Industry 4.0 advocates a completely automatic production line, this inherently also envisions an automatic collection and processing (clustering)of data with no human supervision, and therefore developing a criterion to automatically find this number of classes is of the main benefits in this context.

On the other hand, it is well known that hierarchical clustering does not scale to large datasets. The authors are investigating a novel approach where the hierarchical clustering can be scaled up to large datasets, based on resampling techniques. Details on this research are out of the scope of this paper, but it is relevant to say that the proposed methodology includes steps where automatic identification of the right number of clusters in many dendrograms is more than convenient. The first contribution in this line is [20] wherein the authors proposed a CURE (*Clustering Using Representation*) based [14] strategy with resampling to scale hierarchical clustering for aiding in the identification of the behavioral modes of 3D printers in real production by using sensor data.

Accordingly, this paper presents two novel contributions.

On the one hand, the paper proposes a methodology to automatically evaluate a dendrogram, produced by a hierarchical clustering process, and automatically determine the suitable number of clusters by approaching the way a human expert would have provided. The paper tests several criteria to determine the number of clusters automatically, based on different functions of *Calinski-Harabasz* index [3] to meet human wisdom as much as possible. A comparison with the expert results provides indicators to identify which of the proposed CVI (Cluster Validity Indices) performs more closely to the expert.

As a second contribution, the authors propose the integration of this criterion in a fully automatic methodology based on a modified CURE strategy that considers resampling and is scaling hierarchical clustering for a big data framework that eventually helps in decision making in large datasets. The proposed method and its impact on the time required to process sensor data are illustrated through a real application in the context of 3D printers and processing time is compared with the automatic dendrogram

cut and without with very significant reduction of processing time, as it will be seen.

This study is part of wider research to establish real-time data science methodologies to extract conceptual information from the printing sensors and to connect with an entire IDSS (Intelligent Decision Support Systems) to support the technical management of customer's 3D printers fleet from the manufacturer's side. The presented methodology is general and it is tested with a wide sample from the case study which provides dendrograms of a sufficient variety of topologies to evaluate the validity of the proposal. The methodology proposed in this paper is also potentially applicable to any hierarchical application in general, and especially to other industry 4.0 contexts like gas turbine's monitoring, or aerogenerators monitoring, among others.

The rest of the paper is structured as follows. A brief introduction to the multi-jet fusion 3D printing process is provided in Sect. 2, followed by the survey of available literature in this field in Sect. 3. A detailed methodology of this research is presented in Sect. 4 including a short recap of the previous work in Sect. 4.1. The results on the real dataset are then described in Sect. 5. The strategy to determine the number of clusters automatically is explained in Sect. 6. In Sect. 7, the authors discuss the research contributions and limitations of the proposed methodology and indicate the future scope of the research. Finally, the authors conclude the paper with the results and future scope of the experiments in Sect. 8.

2 Brief introduction to 3D printing process

Additive manufacturing, also known as, 3D (*3-Dimensional*) printing [26, 36] reinvent the manufacturing principles by allowing highly customized parts with substantially less wastage, time, and effort. All of this is achieved by due diligence of the machine with the help of several sensors controlling the operation. This research work has been carried out on *Multijet Fusion 3D Printer* developed by HP Inc. [16].

This particular type of 3D printing starts with a digital 3D-CAD (Computer-Aided Design) file that contains print information that is parsed into the machine with the help of special software. The printing solution consists of three devices, namely the *printer* where the actual printing takes place, the *processing-station* which helps in print-material management, and the *build-unit* that helps to move the print-material (also called *powder*) between the other two devices. A schematic picture is shown in Fig. 1.

The printer initiates a build process that lays down a layer of powder on the print platform which is fused into working parts by the movement of two specific subsystems - *carriage* and *recoater*, while detailing and fusing agents fired onto the powder, define how a part is to be formed. The process is described graphically in Fig. 2. Throughout the process, the machine is connected online and is equipped with several sensors that control various activities of different subsystems. These sensors produce machine logs in real-time to provide real-time functioning of its parts. These logs are further processed into structured tables and eventually used in suitable data mining applications [11].

3 Literature review

Determining the optimal number of clusters (denoted by K) in a dataset is one of the major challenges in any clustering exercise. However, the choice of K is quite subjective to the problem at hand, the similarity metric, and the clustering evaluation criteria. Techniques such as *k-means* demand this value of K to be fed into the algorithm before the clustering starts. On the other hand, *Hierarchical Clustering* [18, 35] provides a flexible approach to decide the best grouping among the objects without needing to specify the parameter K as the number of possible clusters. Broadly, these methods can be categorized as *agglomerative* and *divisive* depending on whether the data objects are grouped hierarchically in *bottom-up* or *top-down* approach respectively. The advantage of constructing a dendrogram or cluster-hierarchy is to be able to cut the hierarchy at any given level and obtain the number of clusters accordingly. The choice of where to cut the dendrogram is what results in the overall quality of clusters, and hence this highly relies on human expertise.

In the absence of any ground truth in unsupervised learning, outcomes of a clustering solution are tricky to evaluate.

In [24, 25], Milligan et al. experimented with 30 different criteria to determine the number of clusters in an artificially generated dataset. The experiment found *Calinski-Harabasz* index to be the most consistent among all the criteria considered in the study.

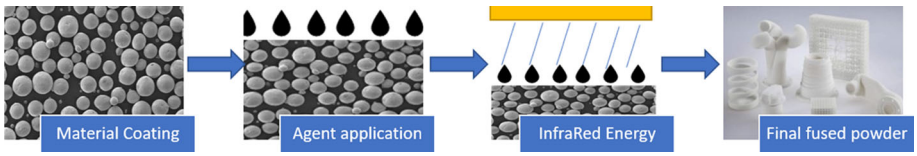
In [34], Tibshirani et al. introduces *Gap-statistic* as a general method to estimate the number of clusters in any clustering algorithm.

The method tests the hypothesis that the null model has a single cluster (K=1) and tries to reject it with an alternative one with ($K > 1$). However, the rejection of the null hypothesis only indicates insufficient evidence in support of the null hypothesis and does not really make it true.

Another common method to search for the best number of clusters is to use *knee* or *elbow* method where an evaluation metric on the Y-axis is assessed against a range of a potential number of clusters on the X-axis and a sharp jump

Fig. 1 HP 3D Printing Solution

Fig. 2 Multi-Jet Fusion Printing Process (©Ashutosh Karna)



is observed as the best clustering solution. The value corresponding to this sharp jump or *knee* is considered as the most suitable value for the number of clusters (K). Sugar et al. in [33] proposed a non-parametric method using *distortion* that computes *Mahalanobis* distance between each multivariate point and the closest cluster center and computes the minimum achievable distortion by fitting different number of clusters. The method visualizes the *jumps* or *elbow* or *knee* in the distortion curve. Salvador et al. in [31] present an L-algorithm that automatically determines the knee in the clustering evaluation curve by finding the boundary between the pair of straight lines that most closely fit the curve, where the evaluation criteria can be chosen as any distance, similarity, or cluster error metric. The methodology, however, needs iterative refinement for a better clustering solution and requires the knee of the curve to be prominently visible to take a judgment. These methods, however, are more suitable to the k-means clustering algorithm where the value of K is often selected by such approaches. The elbow-based methods, however, are not suitable in this case since the authors are dealing with Industry 4.0 sensor data and attempt to discover patterns using unsupervised learning, specifically hierarchical clustering.

In [19], Jung et al. propose *clustering-gain* as a measure for the best number of clusters in hierarchical clustering, where the term *gain* is defined as the difference between the decreased inter-cluster error compared to the initial stage.

In [39] Zhou et al. proposes a new cluster validity index, called CSP (compact-separation-proportion) to obtain a suitable number of clusters. It is an alternative to the CH index which also works with the relationship between the homogeneity-intraclusters and the separability-interclusters, as CH does. However, part of the index is based on the construction of a minimum spanning tree and the proposal is $O(n^3)$ which is prohibitive in large dataset contexts, as our case is.

In [1] and [2], Bruzzese et al. discusses using permutation test to determine the optimal number of clusters in a hierarchical clustering setting, however, this cannot be considered in our large data in quasi-real-time settings.

An automatic cluster selection technique has been discussed in [8] where Ferraretti et al. defines a global value on top of the CVI like Dunn's index [6, 7] Davies-Bouldin, Silhouette, etc. for a non-horizontal cut of the dendrogram. The authors are restricted to horizontal cuts of the dendrograms since the ultra-metric properties of the index level (the vertical coordinates of the dendrogram nodes) are broken when the tree is cut in polygonal forms and this is directly impacting into the heterogeneity of the quantity of information provided by the clusters, and in consequence, on the homogeneous granularity of the knowledge components of the further IDSS based on the discovered clusters.

Yang et al. [37, 38] also proposed a *Hidden Markov Model* based meta clustering approach for temporal datasets and introduced a concept of dendrogram-based similarity partitioning. This paper uses hidden Markov models based hybrid meta-clustering scheme wherein each cluster partition is initially assumed to follow a mixture model whose parameters need to be estimated by minimizing the

loss based on KL-divergence. The max number of clusters are then obtained, which are further finetuned using Hidden-Markov-Model based agglomerative clustering. However, this approach is quite time-expensive in the underlying research context considering the scale of data at hand and the authors are not interested in making any apriori distributional assumptions.

In [4], Cowgill et al. presented a genetic algorithm based approach — *COWCLUS* to optimize the *Calinski-Harabasz* index to determine the right number of clusters. However, the datasets used for the evaluation in the paper, are quite small in size (up to 500 objects) and the time-complexity on larger datasets has not been tested. Another application on meta-heuristics is found in [22] where Liu et al. propose a genetic algorithm based clustering algorithm. This work [22] too has been evaluated only on small datasets (400 objects) and the performance on large scale data is missing, and thereby does not apply to the case study of 3D printers' data.

In [32] several CVI are evaluated on a benchmark of 17 datasets in regards to hierarchical clustering and it is proven that the different CVI assess different topological characteristics of the clusters like the compactness or the diameter, or the quantity of information contained and that the best performances are provided by those indices that evaluate the relationship between the homogeneity-intra-clusters and the distinguishability-interclusters. Among this family, the *Calinski-Harabasz* (CH) index or the Dunn or Dunn-like indices are found. And the CH is the one that is emulated in the visual inspection performed by experts cutting the dendrograms visually.

Provided that the authors intend to insert this automatic cut of the dendrogram in a bigger procedure where processing time is critical, complex modeling like the one proposed in [38] or computational expensive proposals like those from [1] and [2] is not suitable for us.

In [10], Gibert et al. compare the behavior of different clustering metrics on a real heterogeneous data in order to analyze the differences among the cluster results and introduce the software, namely *KLASS*, to assist users in the interpretation of resulting classes to help with knowledge discovery.

The use of post-processing tools and using human wisdom to conceptualize the structure of the cluster that eventually helps in interpreting the scenario better is described in [9]. Related work is also found in [27], which shows a hierarchical clustering based strategy to characterize patients conditions in an Electro-Convolusive-Therapy.

In [13], Gibert et al. analyze the nested partitions and their relationships using special significance tests and hence are related to the research where the authors are dealing with clustering based on partitions. Most of the literature found related to this work is mainly focused on optimizing the number of clusters by some validation criteria; however, the authors are interested in optimizing the number of clusters relative to how human experts visualize and decide to cut the tree.

4 Methodology

The application of this research directly impacts the way data from 3D printers is analyzed and interpreted in the Industry 4.0 context. The authors approach the problem of characterizing different 3D printing scenarios based on the sensor data, in an unsupervised manner. One of the most appropriate data science techniques to get a better understanding of the data is hierarchical clustering, however, it poses a few limitations that must be addressed. Apart from finding the right number of clusters, the other main limitation lies in the sheer size of the data in the case of Industry 4.0 or 3D printers where data is collected almost in real-time on a large scale.

The standard algorithm for hierarchical clustering has a time complexity of $\mathcal{O}(n^2)$ and requires $\Omega(n^2)$ memory, which makes it too slow for large data sets.

The classical hierarchical method is thus not quite suitable for sensor data in Industry 4.0 due to the $O(n^2)$ time complexity of the algorithm which makes it almost infeasible for large datasets.

The proposed methodology is generic and provides a reduction in the computational cost of the algorithm. Although worse case complexity remains the same, the real TCPU time is sensibly reduced, as the quadratic parts of the process are limited to very small subsets of original data and other parts of the algorithm reduce complexity class. The proposal 5 is applied to a real case of 3D printing, although it is suitable for any context with large sensor data. Next, the basics concepts, and the antecedents, together with several aspects of the proposed methodology are presented.

4.1 Previous work

In the previous work [20], the authors proposed a *CURE* [14] based strategy to scale up the computational time of hierarchical-based clustering. The dataset in this study consists of 562,000 records for 41 sensors, collected from eight anonymous HP printers over 300 printing sessions. To deal with the size of the data, a CURE [14] based strategy is used that takes several samples from the large dataset and subject them to local hierarchical clustering. Consequently, a large number of small hierarchical clustering processes are relatively cheaper from the

computational point of view, and thus CURE provides a final step where the clusters are grown with all the remaining objects not included in the samples initially. The proposal provided a quick tool to show the dendrograms to the human-experts to analyze and determine the number of clusters. This was an intermediate step of the proposal that uses human timings. The modified CURE scheme can further be scaled up when the bootstrapping is introduced [21]. The current research work goes a step further and tries to propose a way to determine the cut of the dendrogram automa-tically, by emulating closest to the way how experts do.

4.2 Dendrogram

The dendrogram provides a graphical representation of the hierarchical clustering process. The horizontal axis represents the objects that have been part of the clustering process. The internal nodes of the tree represent intermediate clusters built along the clustering process and the vertical axis (the height of each node of the dendrogram) represents the *level-index* of the clusters which is a measure of the heterogeneity of the cluster itself. Often this vertical axis is related to the inertia of the cluster and when the distance used for the clustering (Euclidean distance in our case, as all variables are numerical) is a metric, the dendrogram itself has an ultra-metric structure, and this guarantees that the subsequent aggregations increase the level index monotonically. Thus, the dendrogram is a representation of the inner structure of the dataset in terms of the inertia of subgroups of objects.

Any horizontal cut of the dendrogram produces a partition of the original dataset in a group of clusters. The horizontal cuts in the higher levels of the tree provide a set of highly heterogeneous classes that correspond to generic classes. While near the bottom of the tree, the horizontal cuts provide a large set of classes that have fewer objects and are more homogeneous (corresponding to highly specific classes). Experts have to find a trade-off between a sufficient number of clusters that result in informative classes and are small enough that they can differentiate among them, while the level of generality must correspond well to the goals of the analysis. In any case, the good place to cut a dendrogram is where a disruption on the heterogeneity of the classes is evident and this corresponds to the place in the tree with gaps in two successive aggregations and with long branches. While inspecting the dendrogram, the human experts carefully examine the relationships among the height of the nodes and identify the biggest gaps as the level below which the heterogeneity of clusters is significantly different than that at the level above this.

The authors in this research work attempt to mimic the process followed by an expert cutting the dendrogram, by using the *Calinski-Harabasz* index, as traditionally this is the cluster validity index that is more related with the decomposition of the inertia in the clusters and presumably, the one measuring most similar to what experts use in their intellectual process.

4.3 Problem formalization

As the research work focuses on sensor data from 3D printers, all the attributes are numerical in nature, and hence Euclidean distance is appropriate as a distance measure and used further in the clustering exercise. Considering that the multivariate data is represented as I, comprising of N multivariate items - $i_1, i_2, ...i_N$, the primary goal in hierarchical clustering is to achieve a sequence of nested partitions $P_K, K = 2, 3, ...N - 1$

$$P_K = \{C_1, C_2, ...C_K\}; k = 1, 2, ...K \quad (1)$$

where C_k represents the kth cluster of the P_K partition of I. The successive P_K are composed of disjoint clusters covering I. Thus, the dendrogram maps into the sequence $P_2, P_2, ..., P_{N-1}$ and $\forall K \in (2, 3, ...N - 1), P_K$ is nested in P_{K-1} so that one of the clusters of the P_{K-1} subdivides in two in the P_K.

The objective is to develop an automatic criterion to identify which of the P_K partitions is more appropriate and matches the most with the expert solution.

4.4 Cluster validity indices

Determining the quality of a given P_K is difficult in real applications as clustering is inherently a non-supervised method, and thus there are no actual labels of items in I to be used as ground truth to compare with. Hence, the authors rather measure the quality of the clusters with some metrics that evaluate the structural properties of P_K. Gibert et al. [10] provides a comparison among several clustering validity indices. Some of the popular cluster validation indices in common practices are as follows.

- *Davies-Bouldin* index [5]: Computes the maximum interclass to cross-interclass distance ratio
- *Dunn's* index [7]: Computes the ratio between the smallest cluster distance and the largest intra-cluster in a partitioning.
- *Silhouette* index [28]: Computes ratio of maximum class spread to variance to determine how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

- *Calinski-Harabasz* index [3]: Computes ratio of *between-cluster* to *within-cluster* inertia adjusted by the number of clusters in the partition.

Amongst several of the cluster validity indices, the *Calinski-Harabasz* index is the one using the decomposition of inertias and which better imitates the process that human experts' follow visually to decide where to cut the dendrogram. Also, there is a correspondence between the maximization of the *Calinski-Harabasz* index and the visual inspection rule used by experts to cut the tree. Therefore, it is decided to use the *Calinski-Harabasz* index [3] in this proposal.

4.5 *Calinski-Harabasz* Index

The *Calinski-Harabasz* index provides a ratio of Between-clusters to Within-clusters inertia of the dataset. Mathematically the index is defined as the following:

$$\mathrm{CH}(K) = \frac{B(K)/(K-1)}{W(K)/(N-1)} \tag{2}$$

where $B(K)$ is *between-cluster* sum of squares, $W(K)$ is *within-cluster* sum of squares, K is the number of clusters and N is the total size of the data.

Also, the terms, $B(K)$ and $W(K)$ are obtained by decomposing the total inertia (or variance) of the data set ($T(I)$) such that:

$$T(I) = B(K) + W(K) \tag{3}$$

Per Huygens theorem, minimizing the within-cluster inertia of a partition (homogeneity within the cluster) is equivalent to maximizing the between-cluster inertia (separation between clusters). In the original work, Calinski and Harabasz [3] suggested choosing a value of K for which the variance-ratio is maximized.

4.6 Proposal of an automatic method to cut a hierarchical tree into a set of clusters

Preliminary experiments conducted by the authors revealed that the original *Calinski-Harabasz* index does not perform well on real data as expected. In fact, in the original paper, the authors discuss about the local maxima that can make a difference. Since the computation of local maxima in discrete spaces is not that easy and the authors are looking for a quicker method to compute these indices, five additional functions are introduced on the *Calinski-Harabasz* index to be optimized over a dendrogram for finding the recommended value of K by global maximization.

As local maxima are associated with null derivative which in turn, is related with differences in discrete scenarios, the authors considering some criteria based on the differences of the *Calinski-Harabasz* index for two consecutive partitions P_K and P_{K-1} and other comparative functions that might find the places where the *Calinski-Harabasz* experiments bring bigger changes along the dendrogram. The 6 criteria are thus described in detail below.

– **M Criterion** The M criterion is the original maximization of *Calinski-Harabasz* index [3].

$$M_K = \mathrm{CH}(K) \tag{4}$$

– Δ **method** It computes the direct difference between successive pairs of the *Calinski-Harabasz* index

$$\Delta_K = \mathrm{CH}_K - \mathrm{CH}_{K+1} \tag{5}$$

– $|\Delta|$ **method** As the *Calinski-Harabasz* indices may not be increasing monotonically and can result in negative differences in the previous method, the $|\Delta|$ consider the absolute differences giving importance to the magnitude of the gap.

$$|\Delta_K| = |\mathrm{CH}_K - \mathrm{CH}_{K+1}| \tag{6}$$

– ρ **method** It computes the relative change in the *Calinski-Harabasz* index by consecutive clusters.

$$\rho_K = \frac{\mathrm{CH}_K}{\mathrm{CH}_{K+1}} \tag{7}$$

– ∂ **method** This method computes the rate of change when the number of partitions is increased by 1 with respect to the *Calinski-Harabasz* index. Mathematically, this can be defined as the ratio of the differential between the *Calinski-Harabasz* indices.

$$\partial_K = \frac{\mathrm{CH}_K - \mathrm{CH}_{K+1}}{\mathrm{CH}_K} \tag{8}$$

– $|\partial|$ **method** Similar to the concept of considering maxima of absolute values of differences as defined in $|\Delta|$ method, the $|\partial|$ method considers the absolute values of rate of change in the *Calinski-Harabasz* indices.

$$|\partial_K| = \frac{|\mathrm{CH}_K - \mathrm{CH}_{K+1}|}{\mathrm{CH}_K} \tag{9}$$

4.7 Determining the recommended number of clusters

For any of the 6 criteria defined before $f \in \{M, \Delta, |\Delta|, \rho, \partial, |\partial|\}$ the best value of K is given by :

$$K^* = \underset{2 \le k \le N-1}{\operatorname{argmax}} f_k \tag{10}$$

Along the experimentation described in Sect. 5, several things were identified about how experts really behave. In fact, there is a rule of thumb by which most of the time, experts ignore the best cut in 2 clusters as the solution appears to be less informative. From the algorithmic point of view, this differs from the expression defined in equation (10).

To better mimic what experts really do, the criteria has been modified as follows:

$$K_2^* = \underset{2 \le k \le N-1}{\text{argmax}}\ (2) f_k \tag{11}$$

where $\text{argmax}(2)$ denotes the second highest maxima.

However, after a few real experiments, is is realized that the second-best cut is only considered by experts when the best cut is in 2 classes and is not always the case. This corresponds to a new formal criterion as defined below:

$$K_{\text{cond}}^* = \begin{cases} K_2^* & \text{if } K^* = 2 \\ K^* & \text{Otherwise} \end{cases} \tag{12}$$

The results of these methods along with the winning criteria are described in Sect. 5.

4.8 Experimental methodology

Once the different methods to obtain the maxima in the *Calinski-Harabasz* indices are defined, their performance would be assessed on a real dataset. The strategy to identify the most appropriate criterion in the current context is as follows:

I Take S samples of data and get the number of clusters determined by human experts from the dendrogram.

II for each sample, apply a hierarchical clustering method and obtain the corresponding dendrograms τ_S

III For each dendrogram obtain the horizontal cut in K clusters ($K \in 2, 3, .., K_{\max} - 1$) and compute the 5 criteria on each one of those partitions.

Let K_s^* be the number of clusters determined by a human expert for sample s.

IV Apply the automatic criterion to determine the K* for each of the samples, according to each of the proposed criteria f, $f \in \{M, \Delta, |\Delta|, \rho, \partial, |\partial|\}$ Let $K_{f,s}^*$ be the best number of clusters for s^{th} sample obtained by applying criterion f.

V Compare the proposed result with the one given by experts. Here two different metrics are used:

(a) Mean square error between the number of clusters proposed by each criteria along the S samples and the number of clusters proposed by the expert

$$\text{MSE} = \frac{1}{S}\sum_{s=1}^{S}(K_{f,s}^* - K_s^*)^2 \tag{13}$$

(b) Tax of correct number of clusters given a tolerance ε

$$T_{f,\varepsilon} = \text{card}\{|K_{f,s}^* - K_s^*| = \varepsilon\}; \quad s \in 1, 2, ..S; \varepsilon \in 0, 1, .., K_{\max} - 1 \tag{14}$$

we will be interested in analyzing these taxes for small values of ε and the best situation is when the samples concentrate in small values of T for small ε and have low presence for T with high ε

(c) Relative tax of correct number of clusters. It is defined as the following:

$$\pi_{f,\varepsilon} = \frac{T_{f,\varepsilon}}{S} \tag{15}$$

4.9 Impact of automatic cut of the trees in real applications

A key advantage of adopting an automatic cut of the dendrogram using the methods described above lies in deploying a solution end to end without any delay. The authors also determined the criteria that better imitates what experts do when proposing the number of clusters of the dendrogram. The winner has been introduced in the former CURE-based strategy presented in Sect. 5 and implemented in the dataset from the previous work [20].

This criterion helps save time in the process of clustering sensor data coming from 3D printers (or sensor data in *IoT* in general) as it computes quite rapidly than how an expert does. The overall time required to cluster the data is significantly reduced. In Sect. 5, the impact on the total time required to cluster a large dataset is shown by displaying a graph (Fig. 4) with the time comparison before and after introducing the automatic criterion into the loop.

The modified CURE strategy, as presented in [20] can be decomposed into main three phases, from the computational point of view:

I Initialization of the algorithm and drawing of S samples

II for each sample

(a) Perform hierarchical clustering
(b) Determine the number of clusters
(c) Cut the dendrogram

III Super-classification of the centroids of all samples

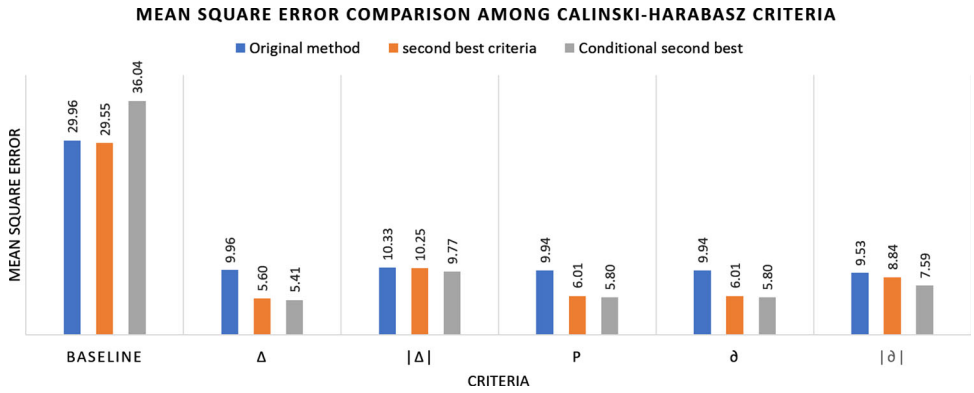


Fig. 3 Mean square error comparison among all criteria

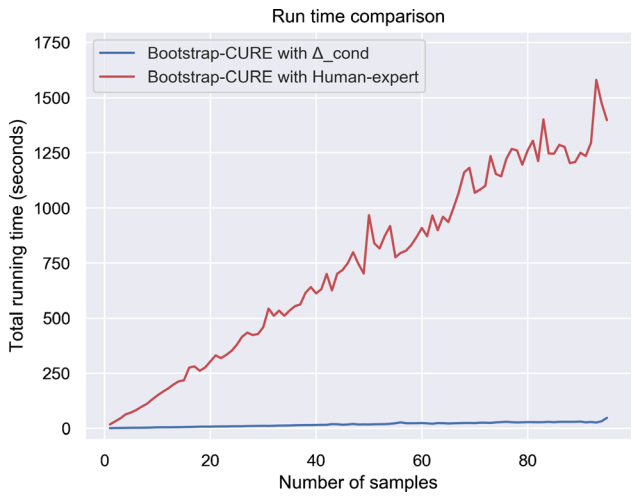


Fig. 4 Comparison between Human-expert and Δ_{cond}

IV Determine the global clustering
V Expand discovered classes to the remaining objects in the dataset

Steps III and IV of the process were originally performed in the following way.

Expert-based identification of number of clusters:

1. Plot the dendrogram.
2. Visual Inspection of the dendrogram by experts and determining the number of clusters.

In this paper the authors propose to substitute this step by an automatic identification of number of clusters:

1. Automatic computation of $P_1{:}P_K$
2. Compute $K_{\Delta^*_{\text{cond}}}$

The authors conducted an experiment to compare the CPU time required under both scenarios. Time for phase 3 is not considered as it is the same in both cases.

5 Experimental analysis and results

In the current research work, the authors are attempting to develop a methodology to determine the number of clusters quickly and as close as possible as humans do, so that they can be introduced in automatic processing for a large dataset, especially in the context of Industry 4.0 applications.

In order to evaluate the criteria described in Sect. 4, authors have considered the original real-world sensor dataset from 3D printers used in [20], composed of 562,000 observations of 41 sensors each. A total of 95 random samples, containing 10 variables and 500 records, have been drawn from the original dataset without replacement. For each sample, hierarchical clustering using Ward's method and normalized *Euclidean* distance is performed and corresponding dendrograms are drawn.

The *Calinski-Harabasz* index or horizontal cuts of the dendrograms between 2 and 8 clusters are computed. Cuts of the dendrograms over 8 clusters are not considered due to the well-known cognitive limitation of the human brain [23, 30] in efficiently processing categories over 7. Further to this, the clustering is desired to aid in decision support and thus it is needed to stay near to the conceptualizations of the reality that an expert can properly integrate from the cognitive point of view.

The Table 1 summarizes the behaviour of $T_{f,\varepsilon}$ for all 95 samples, as defined in Eq. (14). The value in the first column (*Correct Classification*) of the Table 1 provides the number of samples where the automated method matches the number of clusters provided by the expert for all the basic criteria we defined when the decision rule is K^*. The rest of the columns provide the number of samples with increasing ε between 1:7).

The mean square error against each of the methods is shown in Table 2.

Authors have observed significant disagreement between the number of clusters provided by the experts and the ones obtained automatically with the proposed criteria.

Table 1 Summary of $T_{f,\varepsilon}$ for the first decision rule (K^*)

f	$T_{f,\varepsilon}$								
	$\varepsilon = 0$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 4$	$\varepsilon = 5$	$\varepsilon = 6$	$\varepsilon = 7$	$\varepsilon = 8$
Baseline	11	14	8	10	24	15	13	0	0
Δ_K	16	21	15	20	17	3	3	0	0
$\lvert\Delta_K\rvert$	12	26	18	16	16	4	3	0	0
ρ_K	18	19	15	20	17	3	3	0	0
∂_K	18	19	15	20	17	3	3	0	0
$\lvert\partial_K\rvert$	14	26	19	14	15	4	3	0	0

Table 2 Mean square error comparison for first decision rule (K^*)

Criteria	Mean Square Error
Baseline(M)	29.96
Δ_M	9.96
$\lvert\Delta_M\rvert$	10.33
ρ_M	9.94
∂_M	9.94
$\lvert\partial_M\rvert$	9.53

Table 4 Mean square error comparison for second decision rule (K_2^*)

Criteria	Mean Square Error
Baseline(M_2)	29.55
Δ_{M_2}	5.60
$\lvert\Delta_{M_2}\rvert$	10.25
ρ_{M_2}	6.01
∂_{M_2}	6.01
$\lvert\partial_{M_2}\rvert$	8.84

Going in-depth with the analysis of wrong results, authors realized that experts implicitly ignored 2-class cuts in most of the cases, taking the second-best cut, following a *de-facto* standard associated with the idea that 2-classes are not informative enough in most of the cases. Thus in order to approach the expert's criteria better, the authors introduce a new decision rule K_2^* as defined in Eq. 11 which corresponds to the second-highest maxima of each criterion. Table 3 summarizes the results obtained with this modified decision rule.

In this case, the mean square error is shown in Table 4 of these variants improve when compared with human values. However, there are still large proportions of samples yielding high ε's in Table 3.

Going further to understand the differences between expert values and those obtained automatically, the authors realize that the experts ignore the best cut of the tree only when it is in 2 classes, thus using an implicit conditional decision rule. Thus the differences between human-expert and computer-assisted number of clusters can be reduced significantly by combining the previous two decision rules. Therefore, the authors introduce a third decision rule based on this condition, named as K^*_{cond} (Eq. 12).

The performance of all the criteria under this new decision rule is provided in Table 5.

Introducing the conditional maxima to determine the value of K, reduced the mismatches between human-expert and automated criteria. The reduction in mean square error is also seen in Table 7. In fact, this criteria is increasing the

Table 3 Summary of $T_{f,\varepsilon}$ for second decision rule (K_2^*)

f	$T_{f,\varepsilon}$							
	$\varepsilon = 0$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 4$	$\varepsilon = 5$	$\varepsilon = 6$	$\varepsilon = 7$
Baseline(M_2)	8	13	5	24	18	9	11	7
Δ_{M_2}	33	24	17	14	3	4	0	0
$\lvert\Delta_{M_2}\rvert$	12	30	21	18	8	5	1	0
ρ_{M_2}	30	27	17	14	3	4	0	0
∂_{M_2}	30	27	17	14	3	4	0	0
$\lvert\partial_{M_2}\rvert$	12	31	16	19	9	4	0	0

Table 5 Summary of $T_{f,\varepsilon}$ for conditional maxima rule (K^*_{cond})

f	$T_{f,\varepsilon}$							
	$\varepsilon=0$	$\varepsilon=1$	$\varepsilon=2$	$\varepsilon=3$	$\varepsilon=4$	$\varepsilon=5$	$\varepsilon=6$	$\varepsilon=7$
$\text{Baseline}_{\text{cond}}$	12	5	5	14	21	17	20	1
$\Delta_{K_{\text{cond}}}$	43	11	16	16	3	4	2	0
$\lvert\Delta_{K_{\text{cond}}}\rvert$	15	30	21	14	8	5	2	0
$\rho_{K_{\text{cond}}}$	42	12	16	16	3	4	2	0
$\partial_{K_{\text{cond}}}$	33	17	18	17	3	5	2	0
$\lvert\partial_{K_{\text{cond}}}\rvert$	17	29	23	11	8	5	2	0

Table 6 Distribution of $\pi_{f,\varepsilon}$ for conditional maxima rule (K^*_{cond})

f	$\pi_{f,\varepsilon}$							
	$\varepsilon=0$	$\varepsilon=1$	$\varepsilon=2$	$\varepsilon=3$	$\varepsilon=4$	$\varepsilon=5$	$\varepsilon=6$	$\varepsilon=7$
$\text{Baseline}_{\text{cond}}$	0.13	0.18	0.23	0.38	0.60	0.78	0.99	1.00
$\Delta_{K_{\text{cond}}}$	0.45	0.57	0.74	0.91	0.94	0.98	1.00	1.00
$\lvert\Delta_{K_{\text{cond}}}\rvert$	0.16	0.47	0.69	0.84	0.93	0.98	1.00	1.00
$\rho_{K_{\text{cond}}}$	0.44	0.57	0.74	0.91	0.94	0.98	1.00	1.00
$\partial_{K_{\text{cond}}}$	0.35	0.53	0.72	0.89	0.93	0.98	1.00	1.00
$\lvert\partial_{K_{\text{cond}}}\rvert$	0.18	0.48	0.73	0.84	0.93	0.98	1.00	1.00

Table 7 Mean square error comparison for conditional maxima rule (K^*_{cond})

Criteria	Mean Square Error
$\text{Baseline}_{\text{cond}}$	36.04
$\Delta_{K_{\text{cond}}}$	5.41
$\lvert\Delta_{K_{\text{cond}}}\rvert$	9.77
$\rho_{K_{\text{cond}}}$	5.80
$\partial_{K_{\text{cond}}}$	5.80
$\lvert\partial_{K_{\text{cond}}}\rvert$	7.59

Table 8 Comparison among different decision rules for $\pi_{f,\varepsilon\in0,1,2}$

Criteria	$\pi_{f,\varepsilon=0}$			$\pi_{f,\varepsilon\in0,1,2}$		
	K^*	K^*_2	K^*_{cond}	K^*	K^*_2	K^*_{cond}
Baseline	0.12	0.08	0.13	0.35	0.27	0.23
Δ	0.17	0.35	0.45	0.55	0.78	0.74
$\lvert\Delta\rvert$	0.13	0.13	0.16	0.59	0.66	0.69
ρ	0.19	0.32	0.44	0.55	0.78	0.74
∂	0.19	0.32	0.35	0.55	0.78	0.72
$\lvert\partial\rvert$	0.15	0.13	0.18	0.62	0.62	0.73

number of samples in column $T_{f,\varepsilon=0}$ of Table 5 while $T_{f,\varepsilon=1}$ and $T_{f,\varepsilon=2}$ decreased. For some criteria these three columns cover about 70% of the samples. The distribution of $\pi_{f,\varepsilon}$ (proportion of classification with respect to 95 samples) is summarized in Table 6.

The experimental results show that the criterion denoted by $\Delta_{K_{\text{cond}}}$ performs the best as per the criteria defined in Table 12, maximizing $\sum_{0\le\varepsilon\le2}\pi_{f,\varepsilon}$.

Figure 3 shows the mean square error of all the criteria under the three decision rules proposed in the paper. It is seen that $\Delta_{K_{\text{cond}}}$ returns the smallest mean square error as compared to others.

In Table 8, the overall comparison among the three decision rules for the correct classification ($\pi_{f,\varepsilon=0}$) as well as the differences upto 2 classes ($\pi_{f,\varepsilon\in0,1,2}$ is presented.

Thus, for further research Δ_{cond} is the criteria that will be considered.

In fact, the Spearman's correlation between Δ_{cond} and the human criterion is 0.6356 and the corresponding correlation test indicates significant correlation (p-val= 1E-11). Also, the two-sample t-test is assessed to compare the two criteria and in this case, a two-sided p-value=0.5785, indicates no significant difference between the number of clusters provided by the experts and the same provided by our proposed Δ_{cond} criterion.

6 Introducing the automatic identification of number of clusters in global strategy

In this section, the impact on the running time of the entire CURE-based process is compared under the human assistance with the one using the proposed automatic Δ_{cond} criterion.

The experiment assumes that a human expert should not take more than 3 seconds per sample to visually inspect the dendrogram and decide the number of clusters.

To maintain uniformity throughout the work, all experiments were conducted using *Python 3.6* on an Intel Core i7 processor running at 2.6 GHz using 32 GB of RAM, running Windows 10 operating system.

Figure 4 shows graphically the running time of the entire CURE-based process both before and after introducing the automated cut of the dendrograms, with regards to the number of samples used in phase 2 of the CURE-based strategy.

Of course, the running time is linear with regards to S (number of samples) and the more samples are considered in phase 2 of the CURE-based strategy, the higher is the running time. This holds in both scenarios, although the different scale of both curves makes the automated process apparently constant.

Using the automated method, one can reduce the CPU time drastically by running the steps sequentially without any delay. In this experiment, the complete automated clustering solution consisting of 562,000 original observations with 41 sensors, and S= 95 with N=500 takes only 27 seconds on average, while the expert-based scenario raises to 30 minutes, considering a really efficient expert that can process up to 95 dendrograms in 3 seconds each without getting tired and thereby introducing delays along the process.

Thus, by switching the cluster-determination method from manual to automated Δ_{cond}, one can gain an increase of roughly 60x in the running time. Additionally, with automated algorithms, one can also achieve an end to end method that can be implemented in Industry 4.0 scenarios.

7 Discussion

This research proposes the introduction of an automatic step to cut a dendrogram after hierarchical clustering to scale up to the big data scenarios. The main contributions presented in this paper are as follows. Firstly, the proposed Δ_{cond} index is found to be the one performing closely to how a human expert determines the number of clusters by visual inspection of a dendrogram. This has been backed with the experimental results, shown in Table 8 and based on 95 dendrograms obtained from real settings and representing a wide variety of dendrograms topologies.

As the proposal of this research is to find an automatic criterion that imitates the way experts obtain the number of clusters of a dendrogram, only *Calinski-Harabasz* based methods, which are linked with the visual procedures to cut the dendrograms, are considered. In the paper 6 criteria using different optimizations based on the *Calinski-Harabasz* index are proposed, from the more simple one of direct maximization to more sophisticated ones where first derivative and absolute values are proposed. Human experts cut the dendrogram by finding the biggest horizontal gap on (larger branches). The experiments provided evidence that the real criterion used by experts is not exactly corresponding by direct maximization of the *Calinski-Harabasz* index, but with a maximization of the first difference of the series (with the exception of the cut in 2 clusters, most of the times avoided by the experts by experience, and out of topological considerations). Δ_{cond} in fact, is including this exceptional treatment of the 2-classes cut. Some discrepancies between expert criteria and Δ_{cond} results appear when the differences are negative, but not always, and authors presume that even better results can be obtained by analyzing more carefully which topological characteristics of the dendrograms are associated with the sign of these differences and finding a further refinement of the Δ_{cond} criterion. But for the purpose of current research, the approach of human-experts results provided by Δ_{cond} is enough and these improvements will be addressed in future research.

Additionally, the list of criteria can further be enriched by adding more complex indices to optimize the *Calinski-Harabasz* index, such as the method of the second-derivative which has been not included in this research or, even other cluster validity indices such as *Silhouette-coefficient*, *Dunn's* index, *Davies-Bouldin* index, etc. However, these indices are out of the scope of this paper, as the authors are mainly focused on finding the function of the *Calinski-Harabasz* index among the 6 proposed that better emulates what an expert does in real cases. The authors plan to provide a complete assessment of the proposed index against the commonly used CVI's as the future work of this research.

On the other hand, this new criterion has been implemented in a wider procedure (published as a previous work) that scales hierarchical clustering to large datasets by introducing resampling techniques that require an automatic cut of the dendrograms. The performance of the proposed methodology in a large data framework has shown a drastic reduction in CPU time (Fig. 4) as the data size increases, as compared to the standard method. Using the proposed method, the overall run-time to process the data with clustering reduces to almost 1/60th of the

standard run-time on average and this makes the proposal quite appropriate to be inserted in the real-time production systems such as the 3D printers case study. This reduction in run-time is made possible by splitting the original dataset into several bootstrap samples of a much smaller size, each of which is targeted by hierarchical clustering individually and in parallel. The proposal, therefore, attempts to reduce both space and time complexity of the standard hierarchical clustering based processes. The choice of the number and size of bootstrap samples is still a hyper-parameter that might be further explored along with its impact on the overall clustering process. In fact, while developing the proposed Bootstrap-CURE with Δ_{cond} index, the choice of two key hyper-parameters is very important. These are, first, the range of partitions to be explored by the Δ_{cond}, and second, the number of bootstrap samples in the CURE approach. Considering the cognitive ability of the human brain (literature in [23, 30] defend that the human brain can integrate up to 7 distinct levels), the scope of the current research is limited to maximum 8 classes. Besides, regarding the range of partitions explored in this study (between 2 to 8 clusters), the authors' experience suggests that usually, the two or three best cuts of the dendrogram reside within this range, and rarely the experts would require a higher number of clusters, linked to a set of too specific patterns in most of the real applications. Authors are aware that this particular range may not be sufficient to cover all types of datasets universally, however, in any case, the range of partitions to be explored by the algorithm is parameterized and can be extended just by modifying the corresponding hyper-parameter in runtime.

Furthermore, the current study is focused on the specific use-case of 3D printers where the data did not suffer from any missing values. The performance of the proposed Δ_{cond} index in the datasets containing a significant amount of missing terms is yet to be studied in detail in the future steps of the research. Regarding the outliers, one of the main advantages of working with hierarchical clustering algorithms is that they are robust to the presence of multivariate outliers and can identify them ordinarily by creating small clusters. This, in fact, is one of the purposes of the analysis and further automatic interpretation of clusters will elucidate when the small cluster collects wrong measurements of the sensor or it is identifying anomalies in the machine performance. In this sense, guidelines provided in [12] will be taken into account.

As said before, the current research fits suitably well into a much bigger research goal of building an intelligent decision support system to manage the customer's fleet of machines by data-driven approaches and in real-time. The proposal strengthens this goal in the case of 3D printers by quickly recognizing the operational patterns of the machine in an unsupervised manner and eventually acts as a building block of future IDSS. Knowing the correct number of clusters in a dataset quickly, especially when one deals with large scale systems (such as the one producing data in real-time), is a major initial step to allow a fully automated intelligent decision-making process. Once, the number of clusters is determined, the subsequent steps would involve strategically identification of suitable machine learning algorithms (both supervised and unsupervised) to understand more about the underlying data patterns through automatic interpretation tools, to detect anomalous behavior, predict failure events in advance, and finally communicate with the customer to make recommendations for a better operation of the machines or for solving punctual problems. It is important to remark that clustering represents an initial non-supervised step to discover behavioral patterns on machine performance, extremely useful at the manufacturer's side, as first of all, no additional information from what customer is doing is available in real additive manufacturing fleet, and specific experimental tagging is not suitable in real settings. But also, cause 3D printers are emerging technologies from which no complete knowledge is available, and the casuistic of real performance scenarios in digital manufacturing is still unknown, so, knowledge-based approaches cannot cover yet the entire casuistic required for Intelligent Decision Support Systems construction.

Although the experiment was conducted on data coming from a 3D printer, the proposed approach can easily be extended to use cases outside 3D printing, such as industrial applications where data is collected at a large scale and is also devoid of labels to protect customer confidentiality. In all such cases, clustering is a preferred method to aid in pattern-recognition and hierarchical clustering, in particular, is very helpful to reveal the inner structure of the dataset as an outcome as well as suggest the possible number of clusters. This is crucial as there is no apriori hypothesis about the number of patterns intrinsically existing in each machine.

Consequently, the direct application of this approach allows managing data coming from the customer's machines without explicitly asking to label the records manually. This is especially useful when the customer data is confidential and the actual information about the job or machine-health is confidential to share, or even when the customer's operation cannot be interrupted and insights are to be built rather quickly. Thus, the applications of such an approach are not merely limited to 3D printers only but a range of industrial use cases such as the ones in the cyber-physical system, digital manufacturing, Gas-turbines, aro-generators performance, air-traffic-control or health monitoring system, etc. that follow the similar real-time sensor data-collection process and data itself is of similar

complexity. With the help of the proposed approach, standard hierarchical clustering can be scaled up to help in pattern-finding. For example, in an IoT-driven health-monitoring system, the proposed approach could help to quickly identify the abnormal health parameters of the person in an unsupervised manner and inform the concerned staff immediately, or decipher a pattern leading to failure or death events. Similarly, in the case of a gas-turbine, such a technique would be able to identify different patterns of sensor data and analyze the operational modes of the machine, the performance of the health status of the machine. In general, the proposed technique when introduced in its full form (using Bootstrap-CURE strategy) effectively reduces the complexity of the computational process to automatically interpret data and allows the customers (or machine operators) to make a decision without disrupting the production cycle.

8 Conclusions and future work

The current research work studies the effect of several criteria based on the*Calinski-Harabasz* index in determining the best number of clusters from a dendrogram. Six different functions of the *Calinski-Harabasz* index have been proposed as quality criteria of the partitions induced from a dendrogram, and three different decision rules have been proposed to combine those criteria for identifying the final number of clusters.

The experiments conducted in the research elicited implicit practices that clustering experts apply when visually inspecting a dendrogram to determine the number of clusters. This is successfully formalized in the K^*_{cond} decision rule, leading to an acceptable criterion (Δ_{cond}) that mimics, reasonably well, the work done by the expert when analyses a dendrogram and proposes a number of clusters. Although there is a scope of improvement, and further research is conducted to analyze morphological characteristics of those dendrograms where ε is bigger, the current performance can be accepted, provided that the entire process is introduced as part of a more complex strategy that recombines the clusters in a further super-classification approach that reduces the impact of the errors.

It is to be noted that [3] proposes an inferential procedure to determine the number of clusters of a dendrogram. However, the proposal assumes statistical distributions for the Calinski-Harabasz index which are really difficult to be satisfied by real-life sensor data. In the given research, the authors find this index to measure quite well the partition of the data that shows a bigger relative distance between the distinguishability-between clusters and the homogeneity-intraclusters. However, the distributional limitations of the index as described before invalidate the inferential part of the proposal made by Calinski-Harabasz. Hence, there is a need to develop a modification of the test in a non-parametric version and the state-of-the-art in this particular case is to transform the test based on Fishers' Permutation tests. Again, this is computationally too expensive to introduce in the real-time IDSS that is being built in our context. The authors are currently working on finding ways to reduce the computational time of a non-parametric version of the inferential process proposed in [3] and its adaptation to the proposed Δ_{cond} index. Besides, there is a possibility to use significance tests introduced by Lebart to check which sensor is significant and in which particular cluster. This, in fact, is used for the automatic interpretation of the patterns as per the proposal in [32] and, in our experimental setting, this would involve 46,740 distinct tests, one per each class (12) and variable (41) for each dendrogram (95) with considerable time demand and the need of multiple comparison corrections, like the Bonferroni method. Although, the authors are working towards developing a criterion to reduce the number of tests and thereby the computation time. This particular work is considered to be out of the scope of this paper and is part of future research.

Finally, the criterion with the best behavior has been introduced in the CURE-based strategy proposed in [20] to scale up a hierarchical clustering process to large datasets. Thus, a resampling step involving part of the dataset is introduced and many small hierarchical-clustering processes are performed and combined in a super-clustering process. Later, further expansion of discovered clusters is done for remaining objects that were not involved in the resampling process. The impact of computing the number of clusters in all samples with automatic criteria on the reduction of the total running time is huge.

This work, hence, proves that it is possible to extend hierarchical clustering to large data by using a combination of CURE strategy with resampling techniques and an automatic criterion to determine the number of clusters of each subsample.

Developing such a technique to apply in the 3D printer context is very important as it opens the door to characterize the behavior of the 3D printers automatically, in order to identify operational modes, among others, and makes it feasible to do this process in real-time. Indeed, the proposed methodology contributes to disclose the complex nature of the 3D printing process, which still has lots of open questions and can benefit significantly from further conceptualization. Furthermore, automatic interpretation of sensor data that identifies operation scenarios can support automatic control actions, machine maintenance operations, etc, all important issues in the emergent context of Industry 4.0.

Further research on refining the Δ_{cond} criterion is ongoing, to improve the performance of the entire method. The research to analyze the relationship between bigger ε and the morphological properties of the dendrogram to identify further improvements on the decision rule is in progress. Also, comparison of the proposed Δ_{cond} index against other cluster validity indices from the literature is ongoing, like Silhouette index, Dunn's index, or CSP and their impact on clustering performance as well as on the total CPU time consumption.

In the future work of this research, the authors also intend to apply the proposed methodology to other use-cases involving sensor data coming from different domains (other than 3D printers) to evaluate the generalizability of the proposal as, gas turbines performance, aerogenerators performance, among others.

The proposed methodology is planned to be integrated into an Intelligent Decision Support System to allow the manufacturer to help manage the customer fleet of the 3D printers. Automatic interpretation of the clusters discovered would be the next phase of the research. It would further be connected with supervised and unsupervised learning models to be able to detect anomalous behavior of the machines to catch any system error or part quality concerns.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Bruzzese D, Vistocco D (2010) Cutting the dendrogram through permutation tests. In: Proceedings of COMPSTAT'2010, pp 847–854
2. Bruzzese D, Vistocco D (2015) Despota: dendrogram slicing through a pemutation test approach. J Classif 32(2):285–304
3. Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat-Theory Methods 3(1):1–27
4. Cowgill MC, Harvey RJ, Watson LT (1999) A genetic algorithm approach to cluster analysis. Comput Math Appl 37(7):99–108
5. Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell **PAMI-1** (2):224–227
6. Dunn J (1974) A graph theoretic analysis of pattern classification via tamura's fuzzy relation. IEEE Trans Syst Man Cybern 3:310–313
7. Dunn JC (1973) A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. J Cybern 3(3):32–57. https://doi.org/10.1080/01969727308546046
8. Ferraretti D, Gamberoni G, Lamma E (2009) Automatic cluster selection using index driven search strategy. In: Congress of the Italian Association for artificial intelligence, Springer, pp 172–181
9. Gibert K, Marti-Puig P, Cusidó J, Solé-Casals J et al (2018) Identifying health status of wind turbines by using self organizing maps and interpretation-oriented post-processing tools. Energies 11(4):723
10. Gibert K, Nonell R, Velarde J, Colillas M (2005) Knowledge discovery with clustering: impact of metrics and reporting phase by using klass. Neural Netw World 15(4):319
11. Gibert K, Sànchez-Marrè M, Codina V (2010) Choosing the right data mining technique: classification of methods and intelligent recommendation. Fifth international Congress on environmental modelling and software
12. Gibert K, Sànchez-Marrè M, Izquierdo J (2016) A survey on pre-processing techniques: relevant issues in the context of environmental data mining. AI Commun 29(6):627–663
13. Gibert K, Sevilla-Villanueva B, Sànchez-Marrè M (2016) The role of significance tests in consistent interpretation of nested partitions. J Comput Appl Math 292:623–633. https://doi.org/10.1016/j.cam.2015.01.031
14. Guha S, Rastogi R, Shim K (1998) Cure: an efficient clustering algorithm for large databases. ACM Sigmod Record 27(2):73–84
15. Hermann M, Pentek T, Otto B (2016) Design principles for industrie 4.0 scenarios. In: 2016 49th Hawaii international conference on system sciences (HICSS), pp 3928–3937. IEEE
16. HP: Hp multi jet fusion technology (2020) Technical whitepaper. https://www8.hp.com/us/en/printers/3d-printers/products/multi-jet-technology.html. Accessed 30 May 2020
17. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv (CSUR) 31(3):264–323
18. Johnson SC (1967) Hierarchical clustering schemes. Psychometrika 32(3):241–254
19. Jung Y, Park H, Du DZ, Drake BL (2003) A decision criterion for the optimal number of clusters in hierarchical clustering. J Glob Optim 25(1):91–111
20. Karna A, Gibert K (2019) Using hierarchical clustering to understand behavior of 3d printer sensors. In: International workshop on self-organizing maps, Springer, pp 150–159
21. Karna A, Gibert K. Bootstrap cure: a novel clustering approach forsensor data. State of the art on sensor data scienceand application to 3d printing industry. Computers in Industry (Submitted)
22. Liu Y, Wu X, Shen Y (2011) Automatic clustering using genetic algorithms. Appl Math comput 218(4):1267–1279
23. Miller GA (1956) The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol Rev 63(2):81
24. Milligan GW (1981) A monte carlo study of thirty internal criterion measures for cluster analysis. Psychometrika 46(2):187–199
25. Milligan GW, Cooper MC (1985) An examination of procedures for determining the number of clusters in a data set. Psychometrika 50(2):159–179
26. Nale SB, Kalbande AG (2015) A review on 3d printing technology. Int J Innov Emerg Res Eng 2(9):2394–5494
27. Rodas J, Gibert K, Rojo JE (2001) Electroshock effects identification using classification based on rules. In: International symposium on medical data analysis, Springer, pp 238–244
28. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
29. Rüßmann M, Lorenz M, Gerbert P, Waldner M, Justus J, Engel P, Harnisch M (2015) Industry 4.0: the future of productivity and growth in manufacturing industries. Boston Consulting Group 9(1):54–89
30. Saaty TL, Ozdemir MS (2003) Why the magic number seven plus or minus two. Math Comput Model 38(3–4):233–244
31. Salvador S, Chan P (2004) Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In:

16th IEEE international conference on tools with artificial intelligence, pp 576–584. IEEE
32. Sevilla-Villanueva B, Gibert K, Sànchez-Marrè M (2016) Using cvi for understanding class topology in unsupervised scenarios. In: Conference of the Spanish association for artificial intelligence, Springer, pp 135–149
33. Sugar CA, James GM (2003) Finding the number of clusters in a dataset: an information-theoretic approach. J Am Stat Assoc 98(463):750–763
34. Tibshirani R, Walther G, Hastie T (2001) Estimating the number of clusters in a data set via the gap statistic. J Royal Stat Soc Ser B (Stat Methodol) 63(2):411–423
35. Ward JH Jr (1963) Hierarchical grouping to optimize an objective function. J Am Stat Assoc 58(301):236–244
36. Wong VK, Hernandez A (2012) A review of additive manufacturing. ISRN Mech Eng 2012:1–10. https://doi.org/10.5402/2012/208760
37. Yang Y, Chen K (2010) Temporal data clustering via weighted clustering ensemble with different representations. IEEE Trans Knowl Data Eng 23(2):307–320
38. Yang Y, Jiang J (2018) Adaptive bi-weighting toward automatic initialization and model selection for hmm-based hybrid meta-clustering ensembles. IEEE Trans Cybern 49(5):1657–1668
39. Zhou S, Xu Z, Liu F (2016) Method for determining the optimal number of clusters based on agglomerative hierarchical clustering. IEEE Trans Neural Netw Learn Syst 28(12):3007–3017