What Makes a Clustering Set Interesting?

Georg Stefan Schlake
Chair of Data Science
University of Hagen
Hagen, Germany
georg.schlake@fernuni-hagen.de

Christian Beecks
Chair of Data Science
University of Hagen
Hagen, Germany
christian.beecks@fernuni-hagen.de

***Abstract*—Clustering is a core operation in machine learning that aims to determine the inherent structure of a database by grouping data objects based on similarity. Though a multitude of different clustering approaches have been proposed, the subjective nature of clustering makes it difficult to identify interesting clusterings that are beneficial for the application at hand. To this end, Automated Clustering methods strive to not solely retrieve a single best clustering, but a set of clusterings with a high degree of interestingness. However, the notion of interestingness differs across application domains and use cases. In this paper, we investigate 6 different interestingness functions—modelling how likely a set of clusters can offer additional insight to a practitioner—in 4 different scenarios, measured by 7 different evaluation functions—evaluating these interestingness functions externally using a known labelling for each dataset. We note that the choice of an interestingness function as an optimization goal has a significant impact on the result, which is why it is important to make an informed decision about a suitable interestingness function.**

***Index Terms*—Interestingness, Automated Clustering, Automated Exploratory Clustering, Clustering, AutoML, Unsupervised Learning**

I. Introduction

Clustering is a fundamental task in Data Science, particularly in Exploratory Data Analysis [1]. Groupings of similar objects in a database offer new insights into the databases structure, enabling practitioners to gain actionable insights based on "good" clusterings. However, finding a "good" clustering is not only difficult but also highly subjective [2], as it dependents on the application-specific use cases. This makes it difficult to find automated clustering solutions helping practitioners with limited background in data analysis, as conventional AutoML solutions cannot be applied directly [3].

Automated Machine Learning (AutoML) [4] can be considered as a technique to democratize machine learning. This is achieved, by shifting the process of finding and fitting an apt model from the user to a sophisticated use of the machine. This way, even users with limited expertise are able to find fitting machine learning models. This is especially well researched for the supervised learning case, where a clear optimization goal exists [4], [5]. However, in unsupervised learning, a goal for optimization is much harder to find due to unlabeled data and the aforementioned subjectivity.

For these reasons, the idea of Automated Exploratory Clustering is to offer an *interesting* set of clusterings instead of a single clustering [6], with the aim of capturing various aspects of interest. This moves the optimization target from a single clustering, to a set of clusterings, which should complement each other well, deliver new insights but not overwhelm the user. For the user, this will change the problem from generating a clustering—needing experience and insight into the process of clustering geneation—to manually evaluating a small preselection of clusterings and finding a "good" clustering. This will be a task much better fitted to the skillset of a domain expert. To the best of our knowledge, no prior work has addressed the problem of determining whether a given set of clusterings is actually interesting. In this paper, we will thus

- identify a number of different usage scenarios in Exploratory Data Analysis, rising different requirements to the interestingness of a clustering set,
- design a number of different functions evaluating the interestingness of clusterings sets to have a clear optimization goal for an AutoML system, and
- establish a number of measures to evaluate the success of the different interestingness functions in fitting clustering sets abiding to the constraints of the usage scenarios.

The rest of the paper is structured as follows: We will start by giving an overview of related work (section II), followed by introducing our methods (section III). Afterward, we describe the experimental setup (section IV) and summarize their results (section V), before we discuss our findings (section VI) and conclude the paper (section VII).

II. Related Work

In recent years, a variety of methods have been proposed in the field of AutoML [4], [5]. While most of these methods focus on supervised cases, there are also a number of works in the field of Automated Clustering. AutoClust [7] and kClusterHub [8] are methods which first select a fitting algorithm based on the dataset and afterwards optimize its parameters to automatically design a fitting method. While these methods do not directly solve the CASH-problem (Combined Algorithm Selection and Hyperparameter Optimization) [9], AutoML4Clust [10] shows the adaptation of CASH for Automated Clustering. As of now, there exists a plethora of methods utilizing this to generate automated clusterings. These methods include AutoCluster [11], cSmartML [12], cSmartML-Glassbox [13], TPE-AutoClust [14] and ML2dac [15]. The common denominator of all these methods is their goal of retrieving the single "best" clustering for the input dataset,

regardless of the subjectivity of the clustering problem [2]. In contrast to this, AutoClues [16] generates a set of clusterings and uses methods from the field of diversification, to retrieve the most interesting set of n clusterings. In similar fashion, [17] and [18] generate a large set of clusterings and select a smaller subset deemed to be interesting. These works do not directly optimize the interestingness of a set of clusterings, but select clusterings from a pool of already computed clusterings. In [6], clustering sets are directly optimized, using an interestingness function as a target. However, there are no reports on the actual usefulness of the retrieved clusterings, making any evaluation on the utilized interestingness function impossible.

There exists plenty of other works using the term interestingness. [19] uses information theoretic measures to construct a fixed size set of clusterings. In the field of interestingness prediction (see [20] for a survey), potential excitement in images is measured. [21] uses the term interestingness to measure the diversity of sets. However, all these notions of interestingness differ and are not directly applicable to our problem.

III. Methods

Like the value of a single clustering, the interestingness of a set of clusterings is highly subjective as well, leading to different use cases preferring different kinds of clustering sets (subsection III-A). Automated Exploratory Clustering is the problem (subsection III-B) of finding the most "interesting" set of clusterings for a dataset provided by a user. Clustering sets and their corresponding clusterings have different properties, which are useful for the evaluation of these sets (subsection III-C). We will finish this section by proposing a number of interestingness functions (subsection III-D) used to measure the interestingness of clustering sets. Interestingness in the sense of this paper denotes the likeliness, that a user can gain some added value or actionable insight from a retrieved clustering set.

As this paper is focussed on the general method, we assume clean datasets and that our used CVI and similarity metrics are sufficient for each scenario. For this reason, we also only test on synthetical datasets. This limits the direct practical usability of our findings.

A. Use Cases

Clustering, especially when used in an exploratory context, is a subjective task [2]. For example, a single dataset generated by a machine can contain multiple clusterings, where one is more fitting for academical users, while other clusterings are better in the field of medicine, business or environmental sciences (Figure 1).

Like the use of a clustering in itself, the use of a clustering set is also depending on the actual scenario. While in some situations, it is very important to find the best possible clustering regardless of the total number of clusterings to be evaluated, where in other situations, it is very hard to make an informed decision on a clustering, so a small set is more desirable. We have identified the following scenarios:

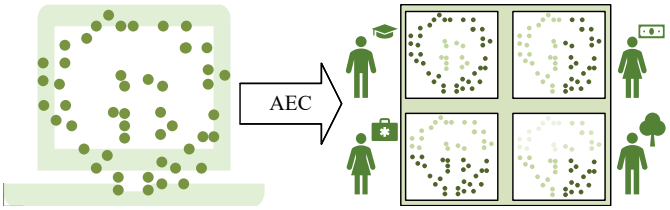


Fig. 1: An example of a dataset with 4 different clusterings, which are useful for 4 different practitioners. All four clusterings seem to be valid in different scenarios. The top left clustering follows a density based approach, whilst the other three clusterings separate the dataset based on spheroid clusters. The upper right and lower left clustering do so based on one dimension each, while the lower right clustering generates four clusters, splitting the dataset in the middle in both dimensions.

Fig. 2: A short overview of the MAS paradigm. A dataset is input to an optimizer loop finding a good solution for the MAS problem, using HPO. The resulting clustering set is returned to the user.

a) Focus on usability.: In this scenario, a user is well capable of evaluating the retrieved clusterings to find the best for their use. The complete focus lies on the quality of the best retrieved clustering, while the size of the clustering set is of minor importance, as all can be evaluated in time.

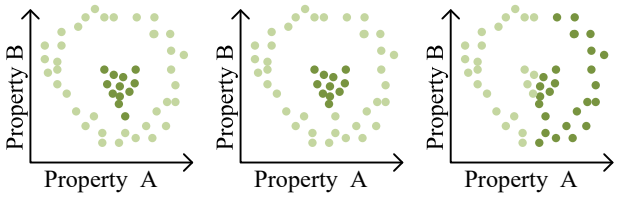
b) Expensive evaluation.: In this scenario, the actual evaluation of a clustering has a high cost, while the usage of a suboptimal clustering has little negative consequences. This leads to the desired clustering set only containing a few clusterings, while the quality of the best clustering is only of secondary nature. This can be useful, if not the "best", but only a "good enough" clustering is needed.

c) Need for many useful clusterings.: In the third scenario, the emphasis is placed on generating as many different views but useful views on the dataset as possible. This is often the case in Exploratory Data Analysis, where a practitioner wants to gain novel insights into a dataset.

d) Need for only useful clusterings.: Another scenario is the search for different clusterings, which however all need to be somewhat useful, as the investigation of each clustering is an expensive and time-consuming process. This can also be useful in Exploratory Data Analysis.

B. Problem Formulation

The problem of Automated Exploratory Clustering (AEC) has been solved using Multi Algorithm Selection (MAS) [6], a slight deviation from the Combined Algorithm Selection and Hyperparameter Optimization (CASH) problem [9].



(a) (b) (c)

Fig. 3: Three different clusterings for the same dataset. Clusterings (a) and (b) are density-based and both split the dataset into a circular and a spheroid cluster, while cluster (c) splits the dataset based on Property A without regard of Property B. Clusterings (a) and (b) differ in the assignment of a single object. While both clustering (a) and (b) seem proper individually, there is no positive effect of using both in a clustering set, as these are not very diverse. A clustering set comprising (a) and (c) would offer more value than a set comprising (a) and (b).

Definition 1 (Multi Algorithm Selection): Given an interestingness function I, an algorithm $A_i \in \tilde{A}$ from a set of algorithms $\tilde{A}$, the set of possible hyperparameters for an algorithm Λ^{A_i}, the search space of all possible configurations $S = \{A_i^\lambda | A_i \in \tilde{A}, \lambda \in \Lambda^{A_i}\}$ and a dataset D, the Multi Algorithm Selection problem can be formulated as follows:

$$R = \underset{\{A_i^\lambda\} \subset S}{\operatorname{argmax}} I\left(\{A_i^\lambda(D)\}\right) = \underset{s \subset S}{\operatorname{argmax}} I(s), \tag{1}$$

where R is the optimum set of algorithms and corresponding hyperparameters subject to interestingness function I.
As can be seen, Figure 1 raises an optimization problem, where the Interestingness function I is optimized on a set of individual clusterings. This is mostly done by using an optimization loop, where an optimizer adapts the hyperparameters of different algorithms to find the clustering set, which grants the best[1] value, until a given computation limit[2] is reached. After this, the best performing clustering set is returned to the user. An overview of this paradigm is depicted in Figure 2. As the choice of the interestingness function has a high impact on the resulting clustering set, we will introduce different interestingness functions in the remainder of this paper and evaluate their capability of fitting to the use cases mentioned above.

C. Useful Properties

Different properties can be used to evaluate the interestingness of a clustering set. On the one hand, each clustering in the clustering set has an intrinsic value, which can be measured by different Clustering Validation Indices (CVI) [22]. On the other hand, clusterings have a value determined by their similarity to other clusterings in the set. As additional clusterings should deliver additional insights, adding a clustering largely identical to an already known clustering will be less useful than a novel clustering with slightly lower CVI values. For example, the clusterings in Figure 3a and Figure 3b both seem to be intuitively good clusterings, but are very similar. If one of these is present in a set, the clustering in Figure 3c would offer more additional insights. Especially if the intuitive assumption that both properties are equally important does not hold, the clustering in Figure 3c might turn out to be of higher value. Hence, it is important to take both quality (measured by a CVI) and usefulness (measured by similarities) into account when evaluating the interestingness of a clustering set.

D. Interestingness Functions

In this subsection, we will define the two interestingness functions introduced in [16] (MMRI) and [6] (MeanANS) as well as 4 novel (MaxANS, MinANS, MinQ, and MeanS) functions, making use of the aforementioned properties. We will denote a clustering set as $C = (c_1, \ldots, c_n)$ where c_i denotes a single clustering, a quality evaluation (based on a CVI) of a clustering c as $q_i(c)$, the set of all quality evaluations of a clustering c as $Q(c) = (q_1(c), \ldots, q_{|Q|}(c))$ and the similarity of two clusterings as $\delta(c_i, c_j)$.

a) MMR-Interestingness.: The first interestingness function is based on Maximal Marginal Relevance [23] and has been successfully used in the field of diversification [24]. It has been introduced to Automated Clustering in [16]. Due to its roots in diversification, MMR is designed to be used with a fixed result size n. For this reason, there is no intricate mechanism controlling the size, as all additional clusterings raise the value of the measure. In the original iterative process of MMR, the clustering c_i which maximizes the following equation based on a previously selected set $\tilde{C}$ is selected:

$$mmr(c_i) = (1-\lambda)Q(c_i) + \frac{\lambda}{|\tilde{C}|} \sum_{c_j \in \tilde{C}} \delta(c_i, c_j) \tag{2}$$

We adopted this formula to get an interestingness function for a clustering set C:

$$MMRI(C) = (1-\lambda) \cdot (|C|-1) \cdot \sum_{c \in C, i \in [1,|Q|]} q_i(c) + \lambda * \sum_{c_i \neq c_j \in C} \delta(c_i, c_j) \tag{3}$$

In this equation, λ balances the tradeoff between quality evaluations and similarities between the clusterings.

b) Interestingness Functions.: Before defining further interestingness functions, we will first define the Average Neighbor Similarity (ANS), which will be used to discourage the similarity among clusterings in clustering set C in the following methods:

$$ANS(C) = \frac{1}{|C|} \sum_{c_i \in C} \max_{c_j \neq c_i \in C} \delta(c_i, c_j) \tag{4}$$

The ANS models the average similarity of each clustering in C to its nearest neighbor. A high ANS will denote a clustering set with a high number of very similar clustering, resulting in a low diversity. A low ANS will show, that all clusterings are

[1] depending on I either maximal or minimal
[2] either time or allowed iterations

significantly different from other clusters, resulting in a diverse and hence interesting set. In the rest of this subsection, we will introduce several interestingness functions based on different aggregations (maximum, minimum and arithmetic mean) of the quality measures on different sets. The Mean CVI by ANS was used in [6]:

$$MeanANS(C) = \underset{c \in C}{\text{mean}}(\text{mean}(Q(c))) - ANS(C) \quad (5)$$

If we only care for the best possible and not all clusterings, we achieve following Interestingness:

$$MaxANS(C) = \max_{c \in C}(\max(Q(c))) - ANS(C) \quad (6)$$

Similarly, we can ensure, that all quality evaluations are of high quality, by only considering the minimal CVI:

$$MinANS(C) = \min_{c \in C}(\min(Q(c))) - ANS(C) \quad (7)$$

A method only using the quality evaluations is to maximize the worst clustering evaluation:

$$MinQ(C) = \min_{c \in C}(\min(Q(C)) \quad (8)$$

Our last method does also not consider the similarities, but only the size of the set against the average CVI:

$$MeanS(C) = \frac{\text{mean}_{c \in C}(\text{mean}(Q(c))}{|C|} \quad (9)$$

In the following sections, we will use the 7 interestingness functions defined in this section, to automatically generate clustering sets and will evaluate their usefulness in the use-case scenarios named earlier.

IV. Experiments

In order to evaluate the capability of the different methods, we designed a typical AutoML loop to optimize clustering sets based on our interestingness functions. We used the optimizer Smac BB [25] to choose up to 25 algorithm instances from the algorithms HDBSCAN [26], [27], *k*Means [28], [29], MeanShift [29], [30], OPTICS [29], [31] and BIRCH [29], [32]. We used the Adjusted Mutual Information (AMI) [29], [33] to measure the similarity of clusterings and the Silhouette Coefficient [29], [34], DBCV [35], [36], VIASCKDE [36], [37] and DSI [36], [38] as quality measures, since these CVI are normalized to [-1,1] [22]. For this reason, they can be compared without any further modification, ensuring that no CVI will have an unproportional influence on the result. As data, we used the 136 synthetic datasets from [39], containing 30 to 10,000 objects in 1 to 40 clusters and two or three dimensions. Our code is available at https://github.com/g-schlake/Interestingness_optimizer.

A. Evaluation Methods

We use both five known methods from [40] (CU, CE, TS, EQC, and QS) and two novel methods (GCS and GQS) catering to our previously identified scenarios to evaluate the performance of the different interestingness functions. This is important, as different scenarios neccesite different clustering sets. According to [40], application cost of a clustering set can be modelled as follows:

$$ac(C, gt, \lambda, U) = \frac{|C|}{U} \cdot \lambda + (1 - \max_{c \in C} \delta(c, gt)) \cdot (1 - \lambda), \quad (10)$$

where gt resembles the original labelling of the dataset and U the maximum possible number of clusterings and a low value resembles a good clustering set. While we do compare with the ground truth of the datasets, we do not consider this ground truth as the *correct* answer for the clustering. However, we can expect this ground truth to be a valid proxy for the best usable clustering. In [40], there are two independent weighting terms, while we model both using a single λ. While we do not believe that gt is necessarily the most interesting clustering in a dataset, we can use this as a proxy. U will have a value of 25 in all our experiments, as this is the maximum possible clustering set size. In previous works, 5 different values for λ have been used for different scenarios. *Costly Usability* (CU) uses $\lambda := 0$, to put the complete emphasis on the usability, disregarding the size of the clustering set (akin to our first scenario), while *Costly Evaluation* (CE) only focusses on the set of the size ($\lambda := 1$). There are also three measures as tradeoffs between these extremes, named *Thorough Search* (TS, $\lambda := 0.25$), *Equal Costs* (EQC, $\lambda := 0.5$) and *Quick Search* (QS, $\lambda := 0.75$). As for our third scenario—generating as many useful but different views as possible—we count the number of clusterings with a similarity to the ground truth above a certain threshold θ as *Good Count Scorer* (GCS):

$$GCS(C, gt, \theta) = |\{c \in C | \delta(c, gt) > \theta\}| \quad (11)$$

We will set $\theta := 0.7$. In order to fit to the range of the other methods, we divide the GCS value by our maximum number of clusterings 25. For our fourth scenario—where a user only cares about useful clusterings—we evaluate the quota of clusterings above a threshold θ as *Good Quota Scorer* (GQS):

$$GQS(C, gt, \theta) = \frac{|\{c \in C | \delta(c, gt) > \theta\}|}{|C|} = \frac{GCS(C, gt, \theta)}{|C|} \quad (12)$$

Like for GCS, we set $\theta := 0.7$ for our experiments, as from our experience, this accounts for clustering deemed similar enough to be relevant. Both GCS and GQS need to be maximized.

V. Results

If we consider our first case, with focus lying on usability, we can see that MMRI seems to deliver the results with the lowest score for CU, with a mean (0.15) even lower than the 25% quartile of all other methods (see Figure 4). MinANS has a significantly higher mean (0.41) than the other methods (around 0.32). The distributions of the other methods are similar to each other, however, MaxANS has a bit better results than the other methods. When looking at the maximum similarity to the ground truth (see Figure 7a), we can see that the clustering sets chosen by MMRI select the sets with the highest AMI compared to the ground truth. If we have a look at our second scenario (and the intermediate measures)

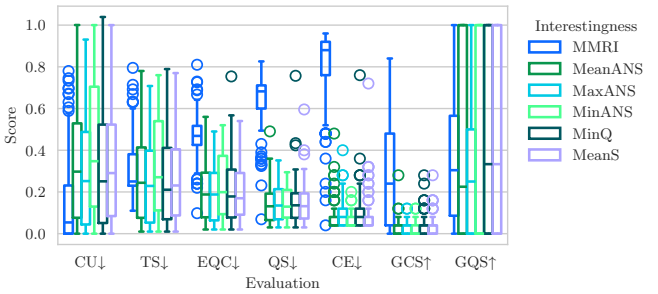


Fig. 4: Evaluation scores of 7 different measurements for clustering sets generated using each interestingness function

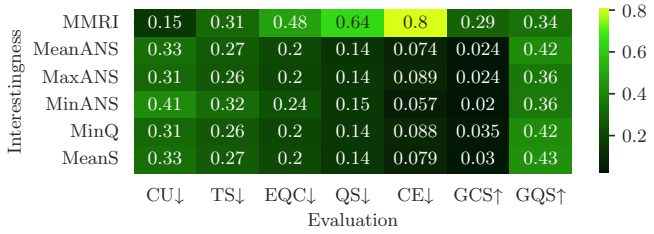


Fig. 5: Mean evaluation scores of 7 different measurements for clustering sets generated using each interestingness function. The arrows indicate, whether each evaluation should be maximized or minimized

this changes, as MMRI selects the biggest clustering sets (see Figure 6), leading to the worst mean QS evaluation of 0.64 and 0.8 for CE. Even using TS, with a still high emphasis on the cluster result compared to the size, MMRI falls behind most other methods and is only better than MinANS. MinANS selects the smallest clustering sizes and has thus the best CE. If we look at the QS values, all methods apart from MMRI seem similar. If we consider the number of good clusterings selected, we can see that MMRI has not only selected the by far most clusterings, but also the most clusterings above our threshold. While all other methods seem to be clearly worse in this regard, MinQ is a small bit better. If we consider the use case, where the quota of clusterings has to be good, we can see that MeanS, MinQ and MeanANS provide the best results. MMRI has the problem of consistently including bad clusterings (see Figure 7b), while the three aforementioned measures in most cases present only clusterings with a similarity to the ground

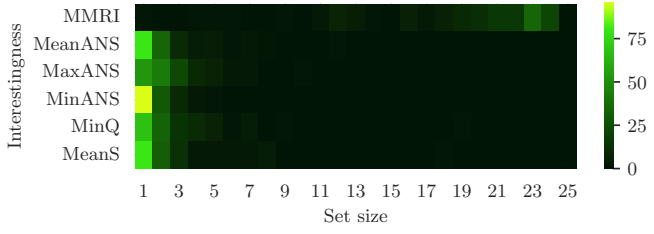
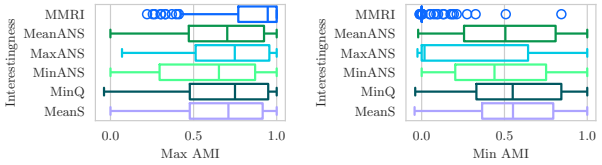


Fig. 6: Number of clustering sets per size



(a) Maximum (b) Minimum

Fig. 7: Maximum and minimum similarities to the ground truth per clustering set

truth higher than 0.5.

VI. Discussion and Future Work

As can be seen, the choice of an interestingness function can have a significant impact on the clustering set generated, making them more or less suitable for different scenarios. The large clustering sets produced by means of MMRI tend to include the most interesting clusterings. However, the other presented methods produce much smaller sets, making them more useful in a couple of scenarios. As there has been little work in this field, all measures are novel, and there is still a wide range of possible interestingness function. Likewise, neither our usage scenarios nor their evaluation are comprehensive, requiring future research in this area, to let users make an informed decision in finding a useful interestingness function for their case.

VII. Conclusion

We have seen, that the choice of an interestingness function significantly influences the resulting clustering sets. Adopting a function from the field of diversification can lead to promising results, given that the size of the clustering set does not negatively influence the usability. If we take this into account, we can see that our novel methods perform better. However, we cannot see a significant difference in the performance of these methods for most scenarios, making it hard to recommend concrete functions.

References

[1] R. C. Dubes and A. K. Jain, "Clustering methodologies in exploratory data analysis," *Adv. Comput.*, vol. 19, pp. 113–228, 1980.

[2] U. von Luxburg, R. C. Williamson, and I. Guyon, "Clustering: Science or art?," in *ICML Unsupervised and Transfer Learning*, vol. 27 of *JMLR Proceedings*, pp. 65–80, JMLR.org, 2012.

[3] G. S. Schlake and C. Beecks, "Towards automated clustering," in *IEEE Big Data*, pp. 6268–6270, IEEE, 2023.

[4] M. Baratchi, C. Wang, S. Limmer, J. N. van Rijn, H. Hoos, T. Bäck, and M. Olhofer, "Automated machine learning: past, present and future," *Artif. Intell. Rev.*, vol. 57, no. 5, p. 122, 2024.

[5] R. Barbudo, S. Ventura, and J. R. Romero, "Eight years of automl: categorisation, review and trends," *Knowl. Inf. Syst.*, vol. 65, no. 12, pp. 5097–5149, 2023.

[6] G. S. Schlake, M. Pernklau, and C. Beecks, "Multi algorithm selection and hyperparameter optimization for automated clustering," in *DSAA*, pp. 1–9, 2025.

[7] Y. Poulakis, C. Doulkeridis, and D. Kyriazis, "Autoclust: A framework for automated clustering based on cluster validity indices," in *ICDM*, pp. 1220–1225, IEEE, 2020.

[8] K. Gratsos, S. Ougiaroglou, and D. Margaris, "kClusterHub: An automl-driven tool for effortless partition-based clustering over varied data types," *Future Internet*, vol. 15, no. 10, p. 341, 2023.

[9] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-weka: combined selection and hyperparameter optimization of classification algorithms," in *KDD*, (New York, NY, USA), pp. 847–855, ACM, 2013.

[10] D. Treder-Tschechlov, M. Fritz, and H. Schwarz, "Automl4clust: Efficient automl for clustering analyses," in *EDBT*, pp. 343–348, OpenProceedings.org, 2021.

[11] Y. Liu, S. Li, and W. Tian, "AutoCluster: Meta-learning based ensemble method for automated unsupervised clustering," in *PAKDD (3)*, vol. 12714, pp. 246–258, Springer, 2021.

[12] R. El Shawi, H. Lekunze, and S. Sakr, "cSmartML: A meta learning-based framework for automated selection and hyperparameter tuning for clustering," in *IEEE BigData*, pp. 1119–1126, IEEE, 2021.

[13] R. El Shawi and S. Sakr, "csmartml-glassbox: Increasing transparency and controllability in automated clustering," in *ICDM (Workshops)*, pp. 47–54, IEEE, 2022.

[14] R. El Shawi and S. Sakr, "TPE-AutoClust: A tree-based pipline ensemble framework for automated clustering," in *ICDM (Workshops)*, pp. 1144–1153, IEEE, 2022.

[15] D. Treder-Tschechlov, M. Fritz, H. Schwarz, and B. Mitschang, "ML2DAC: meta-learning to democratize automl for clustering analysis," *Proc. ACM Manag. Data*, vol. 1, no. 2, pp. 144:1–144:26, 2023.

[16] M. Francia, J. Giovanelli, and M. Golfarelli, "Autoclues: Exploring clustering pipelines via automl and diversification," in *PAKDD (1)*, vol. 14645 of *Lecture Notes in Computer Science*, pp. 246–258, Springer, 2024.

[17] G. S. Schlake, M. Pernklau, and C. Beecks, "Automated exploratory clustering," in *IEEE Big Data*, pp. 5711–5720, IEEE, 2024.

[18] G. S. Schlake and C. Beecks, "The skyline operator to find the needle in the haystack for automated clustering," in *IEEE Big Data*, pp. 6117–6122, IEEE, 2024.

[19] T. De Bie, "Subjectively interesting alternative clusters," in *MultiClust@ECML/PKDD*, vol. 772 of *CEUR Workshops*, pp. 43–54, 2011.

[20] M. G. Constantin, L. Stefan, B. Ionescu, N. Q. K. Duong, C. Demarty, and M. Sjöberg, "Visual interestingness prediction: A benchmark framework and literature review," *Int. J. Comput. Vis.*, vol. 129, no. 5, pp. 1526–1550, 2021.

[21] R. J. Hilderman and H. J. Hamilton, "Evaluation of interestingness measures for ranking discovered knowledge," in *PAKDD*, vol. 2035 of *Lecture Notes in Computer Science*, pp. 247–259, Springer, 2001.

[22] G. S. Schlake and C. Beecks, "Validating arbitrary shaped clusters - A survey," in *DSAA*, pp. 1–12, IEEE, 2024.

[23] J. G. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *SIGIR*, pp. 335–336, ACM, 1998.

[24] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. T. Jr., and V. J. Tsotras, "On query result diversification," in *ICDE*, pp. 1163–1174, IEEE Computer Society, 2011.

[25] M. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, D. Deng, C. Benjamins, T. Ruhkopf, R. Sass, and F. Hutter, "SMAC3: A versatile bayesian optimization package for hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 23, no. 54, pp. 1–9, 2022.

[26] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 1, pp. 5:1–5:51, 2015.

[27] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *J. Open Source Softw.*, vol. 2, no. 11, p. 205, 2017.

[28] J. McQueen, "Some methods for classification and analysis of multivariate observations," in *Berkeley Symp. Math. Stat. Probab.*, pp. 281–297, University of California Press, 1967.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[30] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[31] M. Ankerst, M. M. Breunig, H. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in *SIGMOD*, pp. 49–60, ACM Press, 1999.

[32] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *SIGMOD*, pp. 103–114, ACM Press, 1996.

[33] X. V. Nguyen, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?," in *ICML*, vol. 382 of *ACM International Conference Proceeding Series*, pp. 1073–1080, ACM, 2009.

[34] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.

[35] D. Moulavi, P. A. Jaskowiak, R. J. G. B. Campello, A. Zimek, and J. Sander, "Density-based clustering validation," in *SDM*, pp. 839–847, SIAM, 2014.

[36] G. S. Schlake and C. Beecks, "Arbitrary shaped clustering validation on the test bench," in *DATA*, p. 363–373, INSTICC, SciTePress, 2025.

[37] A. Şenol, "VIASCKDE index: A novel internal cluster validity index for arbitrary-shaped clusters based on the kernel density estimation," *Comput. Intell. Neurosci.*, vol. 2022.1, p. 4059302, 2022.

[38] S. Guan and M. H. Loew, "A distance-based separability measure for internal cluster validation," *Int. J. Artif. Intell. Tools*, vol. 31, no. 7, pp. 2260005:1–2260005:23, 2022.

[39] M. Parmar, "Clustering datasets," 2022. commit: 96df4c32d3d58746c13acf8997449aa1ae54c0e1.

[40] G. S. Schlake, M. Pernklau, and C. Beecks, "Automated exploratory clustering to democratize clustering analysis," *Applied Sciences*, vol. 15, no. 12, p. 6876, 2025.