The reason why cluster validity indices are wrong

Eugen-Richard Ardelean[1,*], Mircea Susca[2], Raluca Laura Portase[1]

[1]Department of Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, Romania

[2]Department of Automation, Technical University of Cluj-Napoca, Cluj-Napoca, Romania

*** Corresponding authors:** ardeleaneugenrichard@gmail.com

ORCID Author IDs:

Eugen-Richard Ardelean: 0000-0002-0098-4228

Mircea Susca: 0000-0001-7409-4124

Raluca Laura Portase: 0000-0002-8985-4728

Abstract: Cluster validity indices, such as Silhouette, Calinski-Harabasz, and Davies-Bouldin, are consistently used to evaluate the performance of newly developed clustering algorithms and to estimate the number of clusters. In this work, we assess the actual validity of these indices.

We propose the Arboris Distance (or the distance of the tree), a minimum spanning-tree approach to computing distance.

Keywords: cluster validity index (CVI); external clustering validation index; internal clustering validation index; clustering performance metric; clustering quality; cluster analysis.

1 Introduction

Clustering is the unsupervised machine learning task of grouping data points based on similarity [1], [2] with applications spanning diverse domains, such as data mining [1], pattern recognition, bioinformatics [2], market analysis [1] and many others. As an inherently unsupervised technique, clustering groups sets of data points without requiring prior knowledge of class labels [1], [3], [4], [5] with the target of maximizing intra-cluster similarity [6] and minimizing inter-cluster similarity [7]. Due to the unsupervised nature of clustering, a significant challenge emerges, specifically determining the quality of the grouping obtained [8] as there is no knowledge of the valid relationships between data points [1]. Clustering has been shown to be an NP-hard problem [9], [10], [11], [12], [13], [14], [15], [16], as a dataset having N data points and K clusters results in K^N possible clustering options [9], [10], [11], [12], [13], [14], [15], [16]. Moreover, the performance of any clustering algorithm can be affected by several factors: dataset size, cluster overlap, cluster imbalance, dimensionality, noise, and number of clusters [2], [17], [18], [19].

The most common option to determine the performance of a clustering algorithm is through Cluster Validity Indices (CVIs) [3], [6], [12], [14], [17], [18], [20], [21], [22], [23]. The evaluation of the results of clustering has received considerable attention along the years and it is still a relevant question today [12], [14], [17], [18]. CVIs attempt to create mathematical models to estimate certain desired cluster attributes, such as cohesion (also called homogeneity, compactness, tightness, connectedness) and separation (also referred to as heterogeneity) [4], [23], [24] with the purpose of measuring how well a certain partitioning of the data reflects the structure of the data [3], [14], [20], [21], [22], [24]. Most CVIs were designed to address certain data characteristics, and as such they are data dependent [19], [24]. Many CVIs [18] have been developed for evaluating the performance of globular clusters; however, not all datasets contain strictly globular shapes and as such they may fail to correctly assess the clustering performance when irregular shapes are present [25], [26], [27]. There are many domains in which non-globular shapes are common, such as geospatial data [28], animal movements [29], cosmology [30], spatial omics [31], social and biological networks [32], and many others. Therefore, the choice of CVI can influence the quality of the clustering through the resulting scores especially when the choice of clustering algorithm or parametrization is based upon it. Moreover, the choice of CVI may even sway the observer towards a worse clustering. Choosing the most suitable CVI for a specific clustering problem is an important and difficult issue that determines the quality of results and their interpretation.

CVIs can be categorized as external or internal [6], [22], [27], [33]. External indices measure how accurately the clustering labels match a predefined set of labels (also known as the ground truth, the true labels, or the reference labels) [34], [35]. This limits their use to datasets that contain a true set of labels [36]. It has been noticed that external CVIs also suffer from limitations, such as favoring higher/lower numbers of clusters [37] and failure on imbalance datasets [38], [39]. In contrast, internal indices measure cluster attributes such as compactness, separation, structure and shape based on the clustering labels without requiring a ground truth [4], [35], [36], [40]. Most internal indices measure cluster attributes based on the concepts [18], [27] of intra-cluster (compactness) and inter-cluster (separability) distances, thus inadvertently evaluating the morphology of the clusters (especially when using the Euclidian distance) resulting in a bias towards the traditional dense well-separated globular clusters [25]. Thus, there may be cases in which the correct clustering may receive lower performance values than a wrong clustering which results in better values for the cluster attributes estimated by the CVI. This statement is especially true when considering non-globular clusters [25], yet it is not limited to this single scenario. As such, several CVIs have been developed to handle non-globular shapes [25], [41], [42], [43], [44], [45], [46], yet their usefulness remains limited due to drawbacks [25]. Although several graph-based CVIs [44], [47] exist, it has been claimed that [25] graphs based on the Euclidean distances without introducing concepts of density still favor globular shapes and the use of cluster centroids is inappropriate for non-globular clusters [25].

One common use-case of internal CVIs is to estimate the number of clusters [4], [24], [27], [48]. Clustering algorithms can be reapplied with different parametrizations (such as the k of K-Means [49]) with the purpose of obtaining different numbers of clusters to evaluate which of these renders the highest score of a chosen CVI [4], [14]. In such a way, the ‘optimal’ number of clusters may

be determined without prior knowledge of the correct number of clusters. Several studies measure the capability of CVIs to estimate the number of clusters [4], [14]. However, a study [4] of 68 internal CVIs on 21 synthetic and real datasets based on the K-means clustering algorithm found that only 14 out of 68 CVIs are actually invariant to the number of clusters [4]. Thus, CVIs are inherently biased by the number of clusters and as such the results obtained are as well, raising the question whether CVIs are actually useful in determining the unknown number of clusters and whether the results obtained are relevant.

Another common use-case is the evaluation of performance [24], [50] for newly-developed clustering algorithms in comparison to other clustering algorithms [14] based on one or more CVIs. Relying on a single CVI is fraught with pitfalls as it has been shown that different CVIs optimize different, and sometimes conflicting, aspects of clustering [22]. Therefore, CVIs may disagree about which clustering partition is "best", raising questions about the validity of the clustering obtained, the claimed superiority of newly-developed clustering algorithms and the capability of CVIs to evaluate performance.

All CVIs have inherent limitations [18], [51] and should be used in conjunction with other indices, noting that domain knowledge, dataset characteristics, the type of clustering algorithm applied and distance metrics can all influence the values obtained. It has been shown that generally CVIs tend to fail their purpose in scenarios including non-globular clusters [18], [20], [22], [27]. There is no golden standard regarding CVIs which outperforms all others in all scenarios [34], [48], [51], [52], [53].

We attempt to convince the reader that most CVIs are not capable of correctly evaluating the performance of clustering, and they do not even assign the highest evaluation score to the true labels biasing the results. As in many other works, we analyze the performance of CVIs across a plethora of clustering benchmark datasets; however, to prove our claims, the usual analysis of performance of CVIs is extended with the concept of labelsets. Thus, each dataset will be assigned six different labelsets including the ground truth labels, random labels and four labelsets obtained through linear separations of the feature space (as given by the following lines: first diagonal, second diagonal, vertical and horizontal midlines). These simple labelsets (besides the ground truth) are plainly wrong for all datasets (excluding very simple cases) and CVIs should easily be able to determine this by assigning a higher score for the ground truth labels than for the other labelsets (preferably a significantly higher score). However, as we shall see, this is not the case for most CVIs.

As the failure of most CVIs on non-globular shapes still remains an issue [25], we propose a new distance metric, called Arboris Distance (AD), based on the concept of the paths in a graph through a minimum spanning tree obtained from a nearest neighbor graph. With this approach, the new distances obtained take into consideration the structure of the data resulting in relative distances instead of the absolute Euclidean distance which is unable to correctly estimate intra-cluster distances for non-globular cluster shapes [40]. Euclidean-based CVIs can be extended to utilize AD allowing for a more accurate evaluation of clustering. Nevertheless, this does not guarantee perfect results and as such we have also created a new CVI based on AD.

In this work, the performance of 28 CVIs was evaluated and compared using both handcrafted labels and labels obtained from commonly used clustering algorithms on 32 synthetic benchmark datasets with a diverse range of characteristics with the purpose of identifying the most appropriate CVI based on the data characteristics and discovering when and why these indices fail. The contributions of this work that distinguish it from prior studies are:

1. Comprehensive comparative analysis: a systematic evaluation of 28 internal CVIs including both traditional and recently developed.
2. Diverse array of scenarios and datasets: the comparative analysis employs both synthetic and real datasets with diverse characteristics, such as cluster overlap, cluster imbalance, noise and high dimensionality. Multiple scenarios are taken into consideration.
3. Unique evaluation scenarios:
    a. Handcrafted labels showcasing simple scenarios of failure
    b. Analysis of performance for validity (breaking points for imbalance and overlap) and for estimating the number of clusters
    c. Aggregated analyses of performance on 32 synthetic clustering benchmark datasets
    d. Statistical validation of the results
4. A new distance metric: based on minimum spanning trees in nearest neighbour graphs
    a. Extension of 3 commonly used internal CVIs
    b. A new internal CVI based upon the distance metric
5. Extension of external CVIs to handle data imbalance with more accurate validation scores.

2 Materials and Methods

2.1 Cluster validity indices/metrics

We specify the terms and notations used in the equations presented in this work in Table 1, let there be a dataset containing N data points divided into K clusters $C_1, C_2, \dots, C_K$, each cluster C_k has n_k points and with the cluster center (i.e. centroid) μ_k, and $d(x_i, x_j)$ denotes the Euclidean distance between points $x_i$and x_j, while in this case μ_l represents the centroid of the closest distinct cluster.

Table 1 - A short description of the meaning of the terms used in the equations.

Term / Notation	**Meaning**
N	The number of points in the dataset
D	The number of dimensions/features in the dataset
$x_i, where\ i \in [1, N]$	A point in the dataset
K	The number of clusters in the dataset

$C_k, where\ k \in [1, K]$	The k-th cluster
$n_k, where\ k \in [1, K]$	The number of points in cluster k
$\mu_k, where\ k \in [1, K]$	The cluster center/centroid of cluster k
$d(x_i, x_j), where\ i, j \in [1, N]$	The Euclidean distance between points $x_i$and x_j

A high number of internal CVIs have been developed throughout the years [18], [22], [23] and several studies and comparative analyses [17], [18], [22], [23], [54], [55], [56] have attempted to rank these indices across various domains and clustering in general. In Table 2, we present the subset of 28 internal CVIs that have been used in this work along with their time complexity, a short description of inner mechanisms, and the possible range of values from worst to best.

Table 2 – Descriptive table of internal CVIs.

Name	**Abbreviation**	**Complexity**	**Description**	**Range [worst, best]**	**Reference**	**Code reference**
Davies-Bouldin index	DB (↓)	$O(ND + K^2D)$	Average similarity ratio of each cluster with its most similar cluster, where similarity is defined as the sum of within-cluster scatter relative to between-cluster separation.	(+inf, 0]	[3], [57], [58]	[59]
Calinski-Harabasz index	CH (↑)	$O(ND)$	Ratio of between-cluster dispersion to within-cluster dispersion, normalized by the number of clusters and total points.	[0, +inf)	[60], [61]	[59]
Silhouette index	S (↑)	$O(N^2)$	Average across all data points for the normalized difference between its mean intra-cluster distance and lowest mean inter-cluster distance	[-1, +1]	[60], [62]	[59]
Ball Hall index	BH (↓)	$O(ND)$	Mean within-cluster variance (average squared distance of points to their cluster centroids), normalized by the number of clusters.	(+inf, 0]	[63], [64]	[65]
Xie-Beni index	XB (↓)	$O(ND)$	Total (fuzzy) intra-cluster scatter divided by the squared minimum inter-centroid distance.	(+inf, 0]	[66]	[67]

Xie-Beni* index	XB*(↓)	$O(ND)$	Variant of Xie-Beni that modifies the separation term (e.g., average or normalized inter-centroid distance) to reduce bias.	(+inf, 0]	[52]	[68]
Dunn index	D (↑)	$O(N^2)$	Minimum pairwise inter-cluster distance divided by the maximum intra-cluster diameter	[0, +inf)	[69], [70]	[65]
Sum of squared errors	SSE (↓)	$O(NKD)$	Sum of squared distances of all points to their assigned cluster centroids (total within-cluster sum of squares).	(+inf, 0]	[71]	[65]
Duda Hart index	DH (↓)	$O(N^2D)$	Ratio between the average pairwise distance within clusters and the average pairwise distance between clusters	(+inf, 0]	[72]	[65]
Beale Index	B (↓)	$O(ND + NlogN)$	Ratio of the within-cluster sum of squares to the between-cluster sum of squares.	(+inf, 0]	[63], [73]	[65]
R-Squared index	RS (↑)	$O(NK)$	Fraction of total variance explained by the clustering	(-inf, 1]	-	[65]
Density-Based Clustering Validity Index	DBCV (↑)	$O(N^2D + N^2logN)$	Density-based cluster validity measure comparing intra-cluster density to inter-cluster density	[-1, +1]	[25]	[65]
Hartigan Index	H (↓)	$O(ND)$	Hartigan criterion measuring change in explained variance when increasing cluster count	(+inf, 0]	[74]	[65]
Centroid-based Silhouette index	cSIL (↑)	$O(N^2)$	Centroid-based silhouette: per-point silhouette computed using centroid distances instead of full pairwise distances, averaged over points	[-1, +1]	[62], [75]	[67]
Generalized Dunn (variant 43) index	GD43 (↑)	$O(N^2)$	Generalized Dunn variant (4/3) using specific choices for inter-cluster distance and intra-cluster diameter.	[0, +inf)	[69], [76]	[67]

Generalized Dunn (variant 53) index	GD53 (↑)	$O(N^2)$	Generalized Dunn variant (5/3) with different distance/diameter definitions	[0, +inf)	[69], [76]	[67]
Partition Separation	PS (↑)	$O(N^2D)$	Aggregate measure of separation between cluster prototypes relative to within-cluster spread	[0, +inf)	[77]	[67]
Renyi's representative Cross Information Potential	rCIP (↓)	$O(N^2D^3)$	Cross-potential between cluster representative distributions	(+inf, 0]	[78]	[67]
Within–Between index	WB (↓)	$O(N^2)$	Combined within-cluster scatter measure (sum-of-squares based)	(+inf, 0]	[79]	[67]
Score Function	SF (↑)	$O(N^2)$	Composite validity score of between-cluster term (measures how far cluster centroids lie from the global centroid), and a within-cluster term (measures average point-to-centroid scatter inside clusters)	[0, +1]	[80]	[68]
SD validity index	SD (↓)	$O(ND)$	Combined scattering (average within-cluster spread) and separation term	(+inf, 0]	[81]	[68]
S_Dbw validity index	SDbw (↓)	$O(ND + K^2D)$	Sum of average within-cluster scatter and density-based between-cluster term;	(+inf, 0]	[82]	[68]
C index	C (↓)	$O(N^2)$	Compares the sum of pairwise within-cluster distances to the theoretically smallest and largest possible such sums for the same partition	[+1, 0]	[83]	-
I index	I (↑)	$O(N^2D)$	Uses compactness and separation to measure how well clusters are separated relative to their internal dispersion	[0, +inf)	[55]	-
CDbw index (Composed Density between	CDbw (↑)	$O(NKD)$	Composite density-aware validity index that combines a compactness (scatter) term with a density-based between-cluster separation	[0, +inf)	[42]	-

and within clusters)			term (density at middle points between clusters).			
Validity Index for Arbitrary-Shaped Clusters based on the Kernel Density Estimation	VIASC KDE (↑)	$O(N^2)$	Per-point compactness and separation are combined and weighted by kernel-density estimates so the index supports arbitrary-shaped and varying-density clusters	[−1,+1]	[46]	-
Compactness Separation Index	CS (↓)	$O(N^2)$	Ratio of aggregated intra-cluster distances to aggregated inter-cluster distances (using maximal within-cluster distances / minimal between-cluster distances).	(+inf, 0]	[41]	-
COP index	COP (↓)	$O(N^2D)$	Measures compactness as average point-to-centroid (or cluster scatter) and separation as the distance to the farthest neighbouring cluster centroid (or similar farthest-neighbor inter-cluster measure).	(+inf, 0]	[84]	-

Although quite old, the S, DB and CH indices are still employed today in the evaluation of clustering performance [56]. In a benchmark of CVIs [23], S demonstrated excellent performance. Moreover, an extensive comparative study has shown that S has the best results [22] and it indicated that some of the best performing clustering metrics are indeed S, CH and DB [22]. Another study found that S obtained the best performance regardless of the amount of overlap, CH had a good performance on linearly separable data, while DB and GD indices had an adequate performance, yet all CVIs performed poorly on non-linearly separable datasets due to the inherent assumptions of data distribution [12]. Another study [17] showed that CH obtained the highest performance, with S, DB and SF being top ranked as well. DB has been found [23] as the most used CVI for meta-heuristic-based clustering algorithms. Whilst another study [55] has found the I index to be a better option than DB and D.

A study [4] of 68 internal CVIs on 21 synthetic and real datasets based on the K-means clustering algorithm found that the RS, H, DB, and COP indices were top-performing with XB and CH having an acceptable evaluation of performance and the S index had the most consistent behaviour across datasets.

CHS does not work well for highly irregular shapes and can be influenced by outliers [18], DBS is sensitive to the parameter choice of the clustering algorithm, to non-Euclidean distance metrics, and to overlapping clusters [18], and SS may not be optimal in scenarios with noise and outliers, it requires a distance metric and for large datasets the distance matrix computation may be expensive [18]. At the same time, the sum of squared errors (SSE) may also be a potential clustering metric, it has been shown to have limited usefulness in non-globular clusters [54].

The DBCV and CDbw indices are among the few internal CVIs that have been developed for density-based clustering solutions [27] being able to handle non-globular cluster shapes. They have been shown to outperform other CVIs in many cases with failures in some scenarios [27]. The DBCV [25] index has been found to outperform [85] other indices such as CDbw [42], DCSI [86], and VIASCKDE [46], for non-convex and density-based (such as those obtained by DBSCAN) clusters. Although all these indices degrade with an increase in noise [85], DBCV has been found to be the most reliable/consistent and it has been shown to not be affected by the spatial arrangement of clusters [85]. DBCV has also been shown [87] to outperform the Silhouette Width, Calinski-Harabasz, Dunn, Maulik-Bandyopadhyay and CDbw indices when considering density-based clustering algorithms.

Needless to say, the conclusion is that there does not seem to be a consensus in the literature.

2.1.1 Edging Distance and its cluster validity indices extensions

Edging Distance (ED) [40] is graph-based approach to computing distances with application to clustering and its validity with the purpose of extending these methods to irregular cluster shapes, presenting three extensions of cluster validity indices and a K-Means extension. Edging distance computes the k nearest neighbours which has a time complexity of $O(ND + NlogN)$. It is based on several iterations which in the worst case can go up to $O(N)$ invoking k nearest neighbours multiple times on unvisited subsets of the dataset, which results in a total time complexity of $O(n^2d + n^2logn)$ in the worst case including the sorting of distances which contributes with a $O(nlogn)$ complexity. Additionally, the edging distance extension of metrics uses the dataset centre computation, which due to the pairwise distance matrix, has a time complexity of $O(N^2D)$.

The Silhouette extension of the edging distance (ED-S) computes cluster means, which in a scenario with balanced clusters would result in $O(\frac{N^2D}{K})$, but in the worst case of having a single big cluster would lead to $O(n^2d)$. ED-S makes ~N*K calls to edging distance which results in final time complexity of $O(KN^3D + KN^3logN)$

The Davies-Bouldin extension of the edging distance (ED-DB) uses the same cluster mean computation. It also computes the inter-cluster distances as the edging distances between all clusters, thus resulting in $O(K^2)$ calls to edging distance resulting in a time complexity of $O(K^2N^2D + K^2N^2logN)$. Finally, the intra cluster distances are computed as the distance between all points in a cluster to their respective mean, thus edging distance is called for each

sample resulting in a time complexity of $O(N^3D + N^3logN)$. In the general case where $K \ll N$, the final time complexity is $O(N^3D + N^3logN)$.

The Calinski-Harabasz extension of the edging distance (ED-CH) as the original version computes all cluster means and the overall mean of the dataset, which has a time complexity of $O(N^2D)$. It computes the edging distance between all clusters means and the overall mean resulting in a time complexity of $O(KN^2D + KN^2logN)$. The intra-cluster distances are computed as the edging distance between each sample and its cluster mean resulting in $O(N^3D + N^3logN)$. Thus, the total time complexity is $O(N^3D + N^3logN)$.

2.1.2 Arboris Distance

We propose a new distance metric to solve the issue of the Euclidean distance for irregular cluster shapes. The Arboris Distance (from Latin *arboris*, meaning "of the tree") is a graph-based distance metric that can capture the topological structure of the data by computing distances using a Minimum Spanning Tree (MST). It can be considered the minimax path distance in the MST. In contrast to the Euclidean distance that computes distances as straight lines, AD follows the structure and connectivity of the data making it suitable for irregular cluster shapes, clusters connected by thin bridges, clusters with varying densities.

AD naturally adapts to varying densities. In dense regions, distances are smaller; in sparse regions, distances reflect the actual connectivity rather than arbitrary straight-line measurements. This has another implication towards noise robustness. By constructing an MST through Prim's algorithm as shown in Algorithm 1, Arboris distance filters out noise and focuses on the core structure as outliers affect only local edges rather than global distance computations. Moreover, as the AD between two points is the maximum edge of the path in the MST, it is sensitive to the largest gap along the path rather than the sum of small noisy differences. This reduces sensitivity to local noise and to variations in density.

For clustering validation, AD better captures intra-cluster cohesion by using the maximum edge along the MST path within a cluster as shown in Algorithm 2 which reflects the worst-case internal separation, whilst the inter-cluster separation is computed through MST paths between clusters revealing the true connectivity, not just centroid distances.

The AD computes a k-nearest neighbours (kNN) graph, where:

$$G = (V, E), V = \{x_1, \dots, x_N), E = \{e_{ij} = (x_i, x_j, w_{ij} = d(x_i, x_j) | x_j \in kNN(x_i)\}$$

Using Prim's algorithm, the MST is computed from the kNN graph:

$$MST = argmin\{\sum d(x_i, x_j) | e_{ij} \in E_{ST} \; where \; E_{ST} \subseteq E \; and \; |E_{ST}| = N - 1\}$$

The AD between data points *i* and *j* is:

$$AD(i,j) = \max\{w_e | e \in Path_{MST}(i,j)\}$$

Metric Properties:

- **Identity**: AD computes distances based on the graph, the distance between a node and itself remains 0:
  - AD(i, i) = 0
- **Symmetry**: AD computes distance using a MST which means there exists a single path from *i* to *j* regardless of direction:
  - AD(i, j) = AD(j, i)
- **Non-negativity**: AD uses the Euclidean distance to compute edge weights in the kNN graph, as such all distances are non-negative:
  - AD(i, j) ≥ 0
- **Triangle inequality:** Let there be a path between nodes *i* and *j* with the max edge $max(w_{ij})$ and another path between nodes *j* and *k* with the max edge of $max(w_{jk})$, as the paths are calculated in an MST, there exists a single path between *i* and *k* that traverses through *j* as well meaning that the distance is the maximum between the two paths. In the worst case, one of the two paths results in a distance of 0 and the equality still holds:
  - $max\_w_{ij} + max\_w_{ik} \geq \max(max\_w_{ij} + max\_w_{ik})$
  - AD(i,j) + AD(j,k) ≥ AD(i,k)

Due to the strengthened triangle inequality, AD can be considered an ultrametric [88].

Algorithm 1 - MST Construction (Prim's Algorithm with k-NN)

```
FUNCTION build_mst(data, n_neighbors)
    visited = boolean array size n (all False)
    edges = empty list
    pq = empty min-heap

    visited[0] = True
    FOR each neighbor in n_neighbors nearest neighbors of point 0:
        push (squared_distance(0, neighbor), 0, neighbor) into pq

    WHILE len(edges) < n-1 and pq not empty:
        (dist_sq, u, v) = pop(pq)
            IF visited[v]: continue
        edges.append( (u, v, sqrt(dist_sq)) )
```

```
    visited[v] = True

    # compute distances from v to all points
    distances_sq = squared_distances(X[v], X)
    distances_sq[visited] = +inf
    pick up to n_neighbors nearest unvisited neighbors
    FOR neighbor in selected:
      IF not visited[neighbor]:
        push (distances_sq[neighbor], v, neighbor) into pq

  RETURN edges
```

Algorithm 2 - Arboris Distance Computation

```
FUNCTION arboris_distance(tree, i, j)
  if i == j: return 0

  # BFS/DFS from i to find path and parent-edge weights
  parent = { i: None }
  parent_edge_weight = {}

  queue = [i]
  WHILE queue:
    node = queue.pop(0)
    IF node == j:
      break
    FOR (nbr, w) in tree.adj[node]:
      IF nbr not in parent:
        parent[nbr] = node
        parent_edge_weight[nbr] = w
        queue.append(nbr)

  # backtrack path j -> i, take max edge
  max_w = 0
  cur = j
  WHILE parent[cur] is not None:
    max_w = max(max_w, parent_edge_weight[cur])
    cur = parent[cur]

  RETURN max_w
```

AD constructs an MST by repeatedly visiting vertices and computing distances from the current vertex to all points to select neighbor candidates; this yields a dominant complexity of $O(N^2D)$

for distance computations plus heap/priority-queue overhead of $O(NK \log N)$. Therefore, the total time complexity of the AD is $O(N^2D + NK \log N)$. The time complexity of a query on the MST to find the path is based on BFS to find the path between two nodes and to extract its maximum edge, is in worst-case $O(n)$ per pair.

Three CVIs have been extended to use AD instead of the Euclidean distance, specifically S, DB and CH. For AD-S, the MST is computed once and the distances between all points to cluster centroids are computed resulting in a time complexity of $O(N^2D + NK \log N + KN)$. For AD-DB, the dominant component is the single MST computation alongside the distance between cluster points and their respective centroids and the centroid-to-centroid distances resulting in a total time complexity of $O(N^2D + NK \log N + KN + K^2N)$. AD-CH, similarly to the original CH, computes the BSS (as the distances between cluster points and their respective centroids) and the WSS (as the distances between centroids and the overall mean of the data) on the MST, summing up to a total time complexity of $O(N^2D + NK \log N + KN)$.

IDEA:

Unlike other metrics, AD-IDEA uses cluster-specific MSTs for intra-cluster distances, capturing the internal structure more precisely, while using the global MST for inter-cluster distances to respect overall data connectivity.

Ratio: Separation-to-compactness ratio

Sum over clusters: Aggregate cluster quality

Range: [0, ∞), higher is better (well-separated, compact clusters)

K cluster MSTs + 1 full MST + K^2 comparisons

The AD-IDEA computes the MST for each cluster taking in total $O(N^2D)$ in the worst case, the same as the full MST, while the centroid-to-centroid distance computations would take $O(K^2N)$ resulting in total time complexity of $O(N^2D + K^2N)$.

$$AD_{IDEA} = \sum_{k=1}^{K} \frac{\max_{x \epsilon C_k} AD(\mu_k, x)}{\min_{k \neq i} AD(\mu_k, \mu_i)}$$

2.2 Datasets

For the analyses presented in this work, 32 synthetic clustering benchmark datasets have been used. Table 3 presents the datasets and their characteristics.

Table 3 – The characteristics of the synthetic benchmark datasets.

Name	**Characteristics**	N	D	K (n_k)	**REF**
a1-3	Varying # cluster	3000/5250/7500	2	20/30/50 (150)	[89], [90]
s1-4	Overlap (increasing overlap)	5000	2	15 (333)	[91]
unbalance	Imbalance	6500	2	8 (3x 2000, 5x 100)	[5]
aggregation	Irregular cluster shapes Overlap	788	2	7	[92]
compound	Irregular cluster shapes Embedded clusters	399	2	6	[93]
d31	Imbalanced Overlapping clusters	3100	2	31	[94]
jain	Imbalanced Semi-moons interlocking	373	2	2	[95]
pathbased	Irregular shapes Overlapping	300	2	3	[96]
spiral	Irregular shapes	312	2	3	[96]
parabolic	Irregular cluster shape (Semi-moons interlocking)	1000	2	2	[97]
ring	Irregular cluster shape (concentric circles)	1000	2	2	[97]
ring noisy	Irregular cluster shape (concentric circles) Noisier clusters	1000	2	2	[97]
ring outliers	Irregular cluster shape	1000	2	2 + 3 outlier clusters	[97]

	(concentric circles) Outlier clusters				
zigzag	Irregular cluster shape	250	2	3	[97]
zigzag noisy	Irregular cluster shape Noisier clusters	300	2	3	[97]
zigzag outliers	Irregular cluster shape Outlier clusters	280	2	3	[97]
trajectories	Irregular cluster shapes	10,000	2	4	Michał Maciąg (Warsaw University of Technology)
x1	Imbalanced clusters	120	2	3	Eliza Kaczorek (Warsaw University of Technology)
x2	Imbalanced Overlapping	120	2	3	Eliza Kaczorek (Warsaw University of Technology)
x3	Imbalanced Overlapping	185	2	4	Eliza Kaczorek (Warsaw University of Technology)
Data1-7	Varying cluster complexities	1000	2	2-5	[40], [59]
r15	Overlapping clusters	600	2	15	[94]
flame	Overlapping clusters Irregular cluster shapes	240	2	2	[98]

lsun	-	400	2	3	[99]
target	Irregular cluster shapes	770	2	3	[99]
twodiamonds	-	800	2	2	[99]
wingnut	-	1016	2	2	[99]
line	-	250	2	2	[97]
dense	Imbalanced clsuters	200	2	2	[97]
fuzzyx	Overlapping clusters	1000	2	5	[97]

Eight real datasets from the UCI data repository have been used in this work, the datasets and their characteristics are presented in Table 4. It has been observed that the UCI datasets do not form sharp clusters [100] and that they are not representative for all problems [100]. It has also been found that DBSCAN provides the lowest performance for many of these datasets [101].

Table 4 – The characteristics of the real datasets.

Dataset Name	**N**	**D**	**K**	**Reference**
ecoli	336	7	8	[102]
glass	214	9	6	[103]
yeast	1484	8	10	[104]
statlog	2310	19	7	[105]
wdbc	569	30	2	[106]
wine	178	13	3	[107]
sonar	208	60	2	[108]
ionosphere	351	34	2	[109]

2.2.1 Labelsets

The reasoning behind the labelsets is that for a CVI to be correct is not enough for it to have a "good" (as in higher/lower depending on the CVI) score for the true labels, but to have the best score for the true labels (as in higher/lower than all other possible labelsets depending on the CVI).

Figure 1 shows the handcrafted labelsets on the ring [97] dataset. These labelsets have been created through linear separation of the feature space. In our analysis, the following labelsets have been used:

- Ground truth labels (TL)
- First Diagonal separated labels (FDL)
- Second Diagonal separated labels (SDL)
- Vertical midline separated labels (VL)

- Horizontal midline separated labels (HL)
- Random labels (RL)

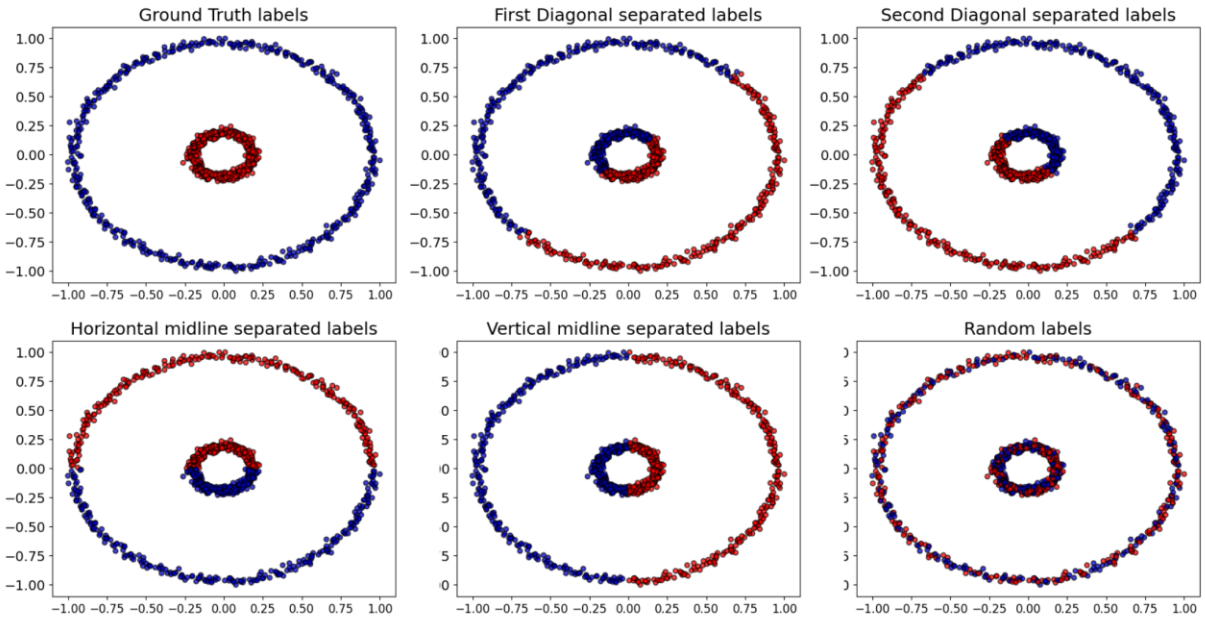

Figure 1 – The six labelsets applied to the ring [97] dataset.

2.3 External cluster validity indices

External CVIs measure how closely the clustering labels match the ground truth labels [34], [35]; thus, they have limited applicability to datasets that contain a true set of labels [36].

The Rand Index (RI) [110] and its extension Adjusted Rand Index (ARI) [111], [112], [113] measure clustering performance as the pairwise comparison of the two sets of labels. Agreements represent cases where two data points are considered to be in the same cluster or in different clusters by both sets of labels, while disagreements represent all other cases. In addition to RI, ARI includes an adjustment for chance agreements. The following equations describe the CVIs, where the expected score *ExpectedRI* represents the score obtained for random labels [112] while the maximum score *MaxRI* is 1:

$$RI = \frac{agreements}{agreements + disagreements}$$

$$ARI = \frac{RI - ExpectedRI}{MaxRI - ExpectedRI}$$

The Mutual Information (MI) and its extension Adjusted Mutual Information (AMI) [113], [114] measure clustering performance as the mutual dependence of two sets of labels U and V. In addition to MI, AMI includes a normalization [113], [115], [116] and an adjustment for chance.

The following equations describe the CVIs where U_x and V_x are two clusters, N is the total number of data points and $|X|$ is the size of a given subset X.

extends the Mutual Information (MI) metric by incorporating entropy (H) into its computation. AMI also incorporates the normalization component of Normalized Mutual Information. It measures the mutual dependence between two clusters and is described by the following equations:

$$MI(U,V) = \sum_{i=0}^{|U|} \sum_{j=0}^{|V|} \frac{|U_i \cap V_j|}{N} log\ log\ \frac{N|U_i \cap V_j|}{|U_i||V_j|}$$

$$AMI = \frac{MI(U,V) - E\big(MI(U,V)\big)}{average\big(H(U), H(V)\big) - E\big(MI(U,V)\big)}$$

2.3.1 Balancing external CVIs

It has been observed in the literature that external CVIs have other limitations as well, such as favoring higher/lower numbers of clusters [37] and failure on imbalanced datasets [38], [39]. We propose an extension to external CVIs that allow them to handle imbalanced datasets.

The proposed approach handles dataset imbalance by turning the global clustering comparison into a set of independent, per-class evaluations and then treating every class as equally important. Instead of scoring the clustering on all samples at once—where large classes contribute many more sample-level comparisons and therefore dominate the final index, drowning errors on smaller classes—it converts each true class into a binary task (in-class vs out-of-class), finds the predicted cluster that best captures that class, computes the chosen external CVI for that binary pair, and records one score per true class as shown in Algorithm 3. Several options are available for the aggregation of per-class scores: macro averaging, weighted averaging, harmonic, etc.

Algorithm 3 – Balancing function for external CVIs

```
FUNCTION BalancedCVI(CVI, true_labels, pred_labels):
    classes ← unique values in true_labels
    IF size(classes) = 1:
        RETURN 1.0

    class_scores ← empty list

    FOR each c IN classes:

        mask ← (true_labels == c)
        true_binary ← mask as {0,1}

        pred_subset ← pred_labels[mask]
        most_common_cluster ← argmax_count(pred_subset)
```

```
    pred_binary ← (pred_labels == most_common_cluster) as {0,1}

    score_c ← CVI(true_binary, pred_binary)
    append score_c TO class_scores

  RETURN mean(class_scores)
```

2.4 Measures

Two of the factors that impact clustering and CVIs the most are imbalance and overlap. To measure the amount of imbalance and overlap in a dataset, the imbalance ratio (IR) and overlap ratio (OR) were used. The IR is computed as the ratio between the count of the largest cluster and the count of the smallest cluster as shown in Algorithm 4.

Algorithm 4 – Imbalance Ratio.

```
FUNCTION imbalance_ratio(X, labels):
  counts = count occurrences of each distinct label in labels

  max_count = maximum value in counts
  min_count = minimum value in counts

  IF min_count == 0:
    RETURN +infinity

  RETURN max_count / min_count
```

The OR computes each cluster's center, and then for every point compares how far it is from its own center to how close it is to the nearest other center; if the nearest other center is within a configurable slack parameter of the point's distance to its own center the point is counted as "overlapping" as shown in Algorithm 5. Finally, the function returns the fraction of points that are overlapping.

Algorithm 5 – Overlap Ratio.

```
FUNCTION overlap_ratio(X, labels, slack = 1.2):
  unique_labels = ordered list of distinct labels
  centers = empty list
  FOR each label IN unique_labels:
    points = subset of X with that label
    center = mean vector of points
    append center TO centers
```

```
N = number of rows in X
overlapping_count = 0

FOR i FROM 1 TO N:
   p = row i of X
   own_idx = index of label of p in unique_labels
   dist_to_own = EuclideanDistance(p, centers[own_idx])

   dist_to_nearest_other = +infinity
   FOR j FROM 1 TO length(centers):
      IF j == own_idx: CONTINUE

      d = EuclideanDistance(p, centers[j])
      IF d < dist_to_nearest_other:
         dist_to_nearest_other = d

   IF dist_to_nearest_other <= slack * dist_to_own:
      overlapping_count = overlapping_count + 1

overlap_rate = overlapping_count / N
RETURN overlap_rate
```

3 Results

3.1 Imbalance and overlap effects on internal CVIs

3.1.1 Imbalance

Let us take a simple dataset and vary the amount of imbalance by regenerating a dataset with 3 clusters varying the number of points of a single cluster (green). We can easily see that the Imbalance Ratio measure computes correctly the amount of imbalance (Figure 2) as the size of minority cluster decreases as the imbalance ratio increases.

We can see that the Silhouette Score is able to correctly evaluate the true labels with a higher score for most cases of imbalance. However, for heavily imbalanced datasets, where close sparse clusters are present, it may have erroneous evaluation as shown in Figure 2 (bottom row) for an Imbalance Ratio of 10.

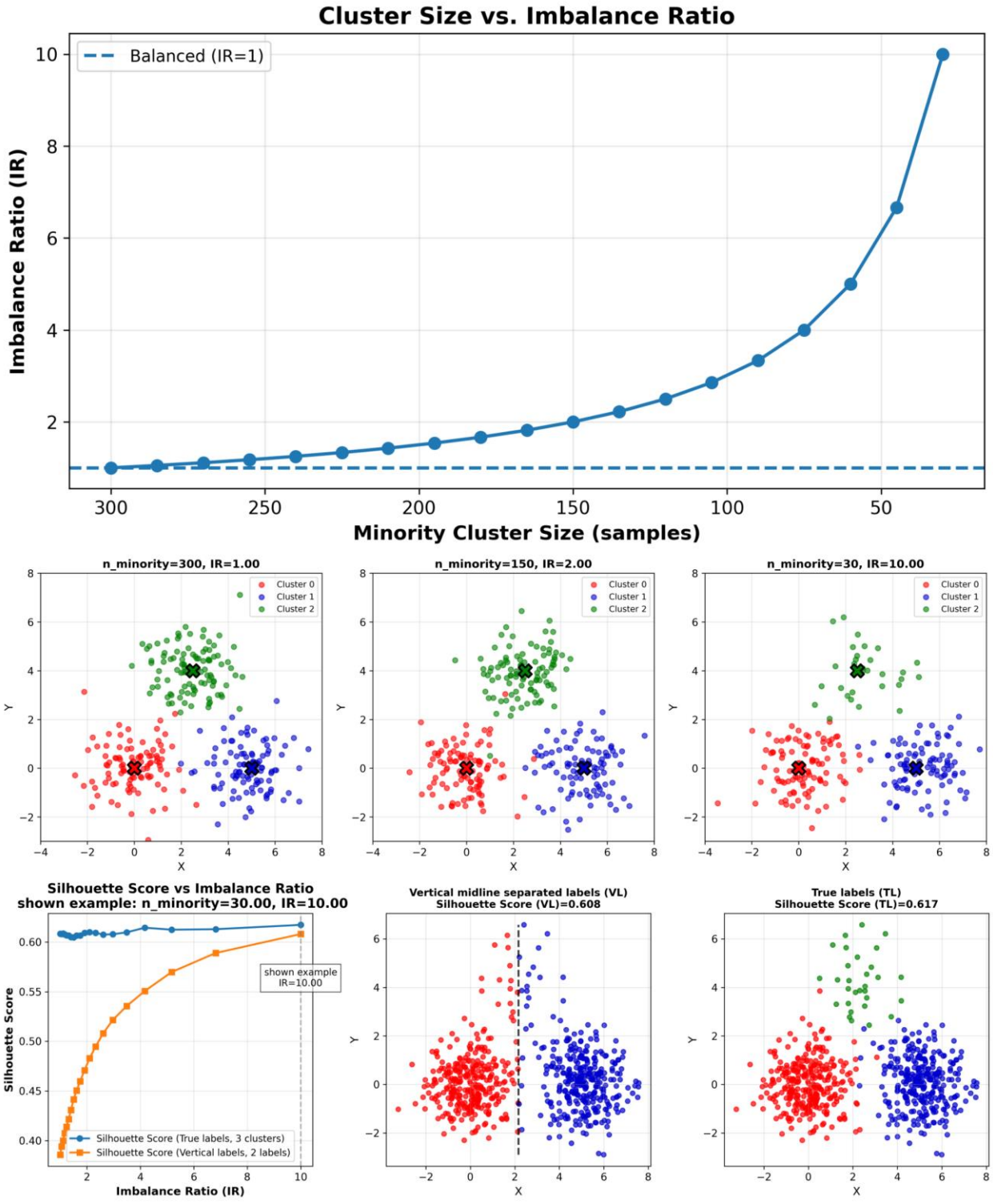

Figure 2 – Impact of imbalance on the S internal CVI.

3.1.2 Overlap

The analysis continues with an analogues scenario for increasing levels of overlap. We can easily see that the Overlap Ratio measure computes correctly (some small variations due to regeneration of clusters) the amount of overlap (Figure 3, middle row) as the distance between clusters decreases, the overlap ratio increases.

In contrast to imbalance, even a small amount of overlap can result in an erroneous evaluation for CVIs as shown in Figure 3, bottom row. Therefore, there is a point at which even though some clustering algorithms would obtain a decent result regardless of the overlap at which a Silhouette analysis would be unable to correctly determine the number of clusters.

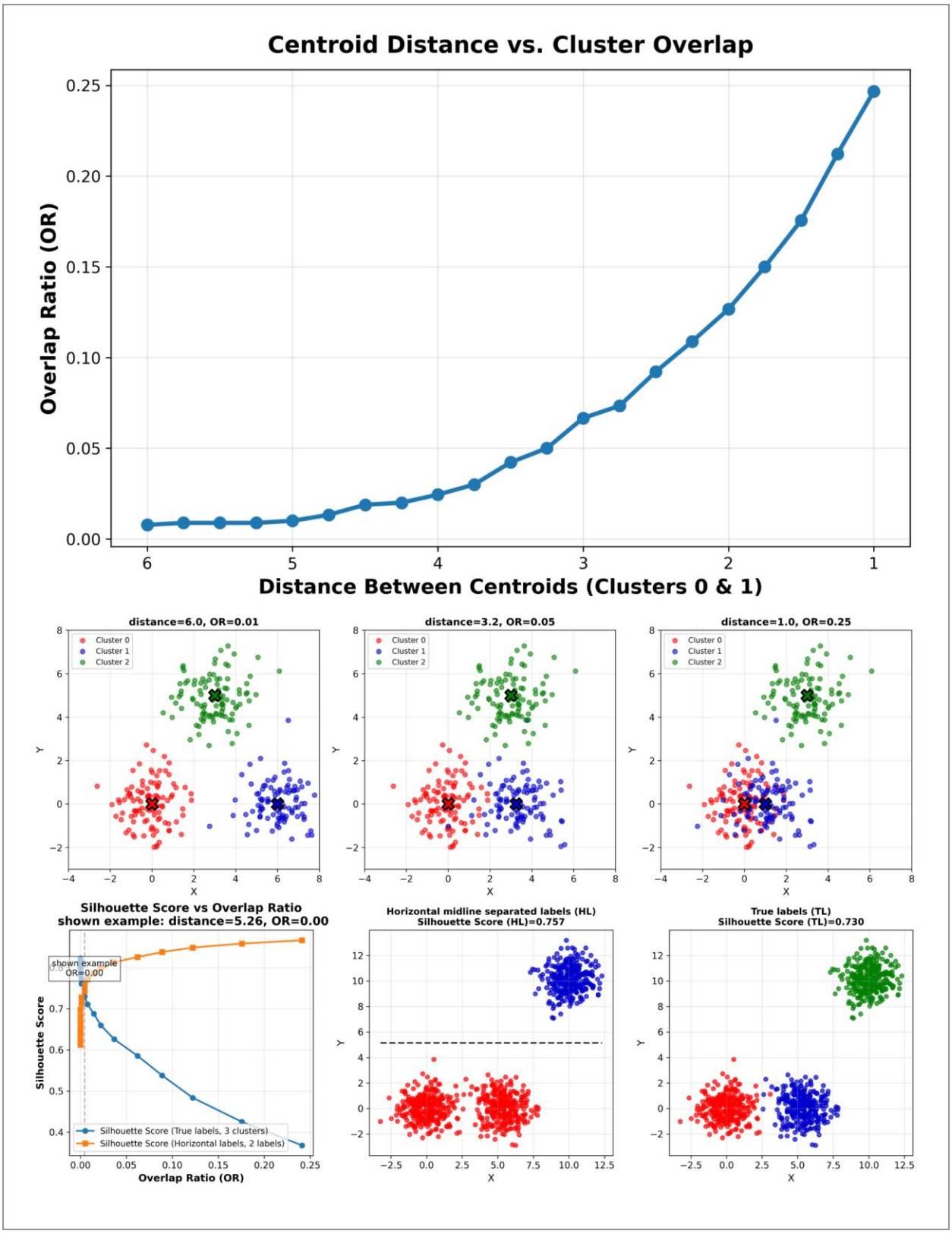

Figure 3 – Impact of overlap on the S internal CVI.

To validate our claim and to demonstrate the validity of the simple yet useful labelsets that we are proposing for analysis, a similar result can be obtained for a K-Means analysis with 2 vs 3 clusters at the exact same amount of overlap shown in Figure 4.

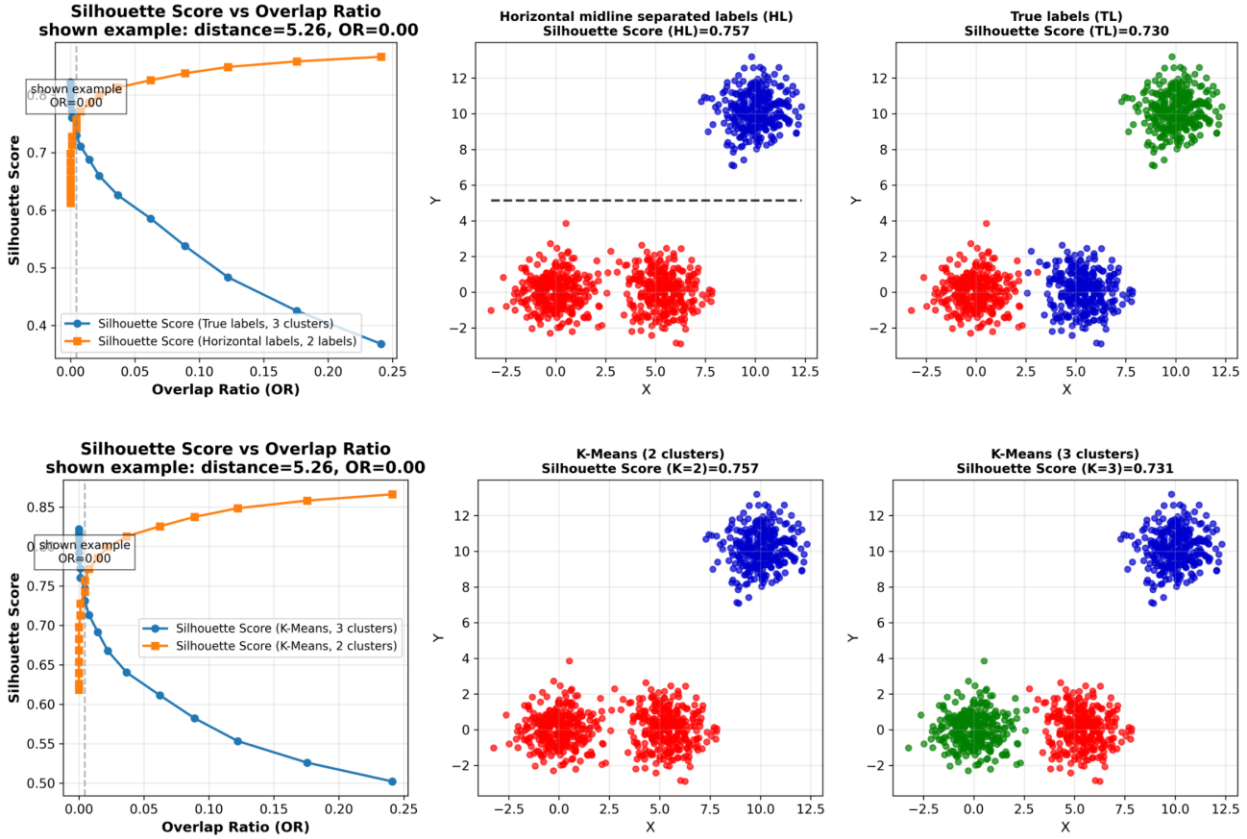

Figure 4 – Comparison between labelsets and different *k* values for K-Means.

3.2 Estimating the 'correct' number of clusters with internal CVIs

This analysis plainly shows that the best partition as defined by a CVI can be incorrect, thus, using the best partition as a reference to validate the performance of CVIs may include even more errors in the evaluation. In this analysis, we have chosen the ground truth as reference with the aim of showing that CVIs are incapable of assigning the highest score to the ground truth. For this analysis, we have chosen the Silhouette index, one of the most commonly used CVIs for estimating the number of clusters.

To demonstrate that this is not an issue of our predefined labels, we also show how actual labels from clustering algorithms perform. It is clear that the ground truth has a significantly lower validation score than either two or three clusters of K-Means (Figure 5) which will be an issue for any clustering algorithm as Silhouette will always "say" that the correct clustering is wrong.

This indicates that regardless of clustering algorithm when searching for parameters whilst looking at the S values, even if a clustering algorithm could find a clustering solution equivalent to the true labels it will not because of the low S score, leading to suboptimal clustering results.

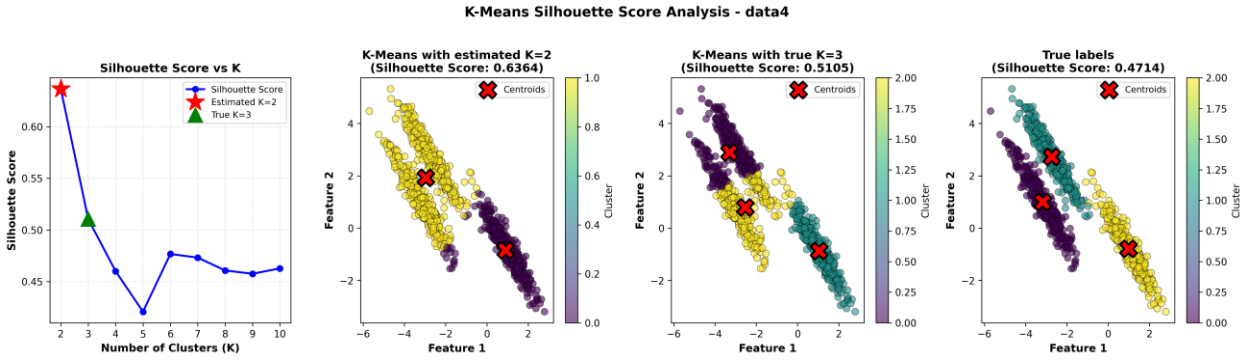

Figure 5 – Estimating the *k* parameter in K-Means using the S internal CVI.

Looking at the AD-Silhouette extension and an analogous analysis in Figure 6, a similar trend is found where K=2 obtains a better result; however, the true labels obtain a higher score than the labels obtained by K-Means for 2/3 clusters.

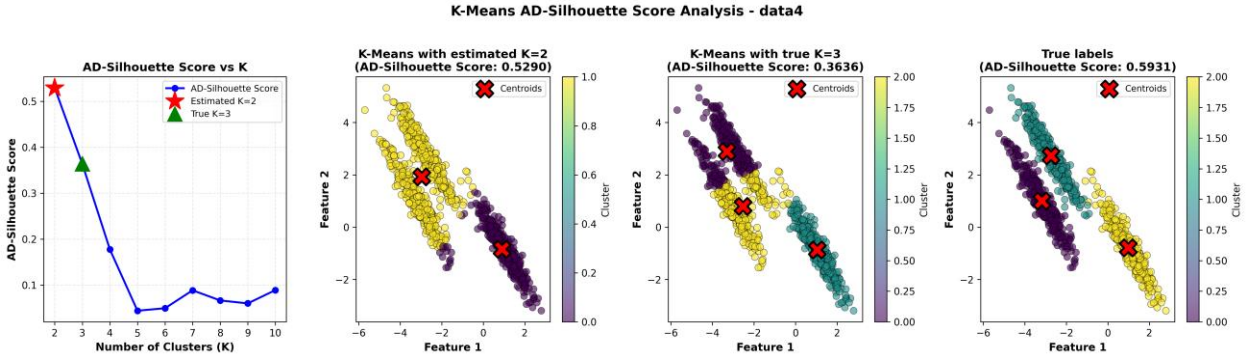

Figure 6 – Estimating the *k* parameter in K-Means using the AD-S internal CVI.

3.3 Errors of external metrics

External CVIs measure how closely the clustering labels match the ground truth labels [34], [35]; thus, they have limited applicability to datasets that contain a true set of labels [36]. However, due to the use of the ground truth, they surely do not suffer from such limitations. In fact, studies show that external CVIs also suffer from limitations, such as favoring higher/lower numbers of clusters [37] and failure on imbalance datasets [38], [39].

Studies [4], [25] have used external metrics to validate the results of internal metrics through correlations or other methods, however due to the erroneous assessments of imbalanced clusters, it raises the question whether these results can be fully taken into account. For this analysis, the

statistics of each cluster of the unbalance [5] dataset were extracted to generate a set of unbalance-like datasets with decreasing amounts of imbalance.

In this analysis, we compare two labelsets against each other, the true labels and a specific labelset with random assignments for the minority clusters (the 5 clusters on the right side with lower number of data points). In Figure 7 (middle row, left), we can see that ARI assigns an almost perfect score with random labels on the clusters with a small count for the ARI. However Balanced ARI (BARI), correctly identifies the imbalance and penalizes through the score value the random labels on the minority clusters. Moreover, if we were to take the dataset and upscale the minority clusters until they have the same number of points as the majority clusters (Figure 7, bottom row), the values of the two scores are almost identical.

In this work, we present the macro averaging, a simple arithmetic mean of per-class scores, in which each class contributes the same amount to the overall metric regardless of how many samples it contains.

Nevertheless, there are trade-offs. Equal weighting makes the metric sensitive to noise in very small classes (a single mislabel can swing that class's score), and choosing only the single most-overlapping predicted cluster may hide situations where a true class is split across several predicted clusters (also known as over-clustering). These are inherent consequences of the local, per-class scoring strategy as it prioritizes per-class fairness over global, sample-frequency-driven accuracy. We recommend the use of this approach when the evaluation is required to reflect per-class performance (for fairness, rare-class detection, or importance of small groups).

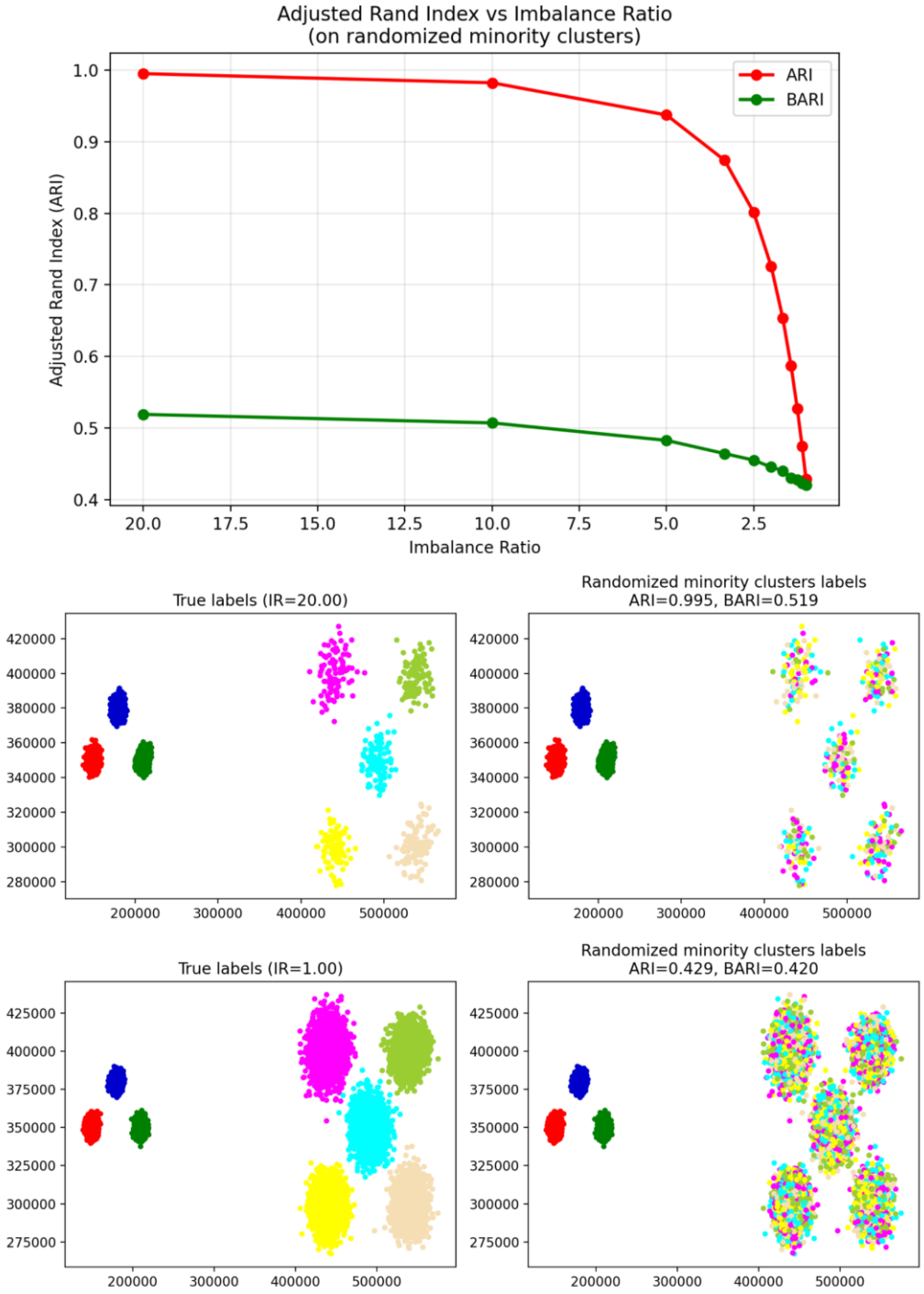

Figure 7. Comparison of external CVIs: ARI and Balanced ARI (BARI).

3.4 Original CVIs vs Edging Distance extensions vs Arboris Distance extensions

In this first analysis, the validity of the Arboris Distance (AD-) extended CVIs is shown in comparison to the original CVIs and the Edging Distance (ED-) extensions on the ring [97] dataset. Table 5 clearly shows that for a dataset with irregular cluster shapes (concentric circles for the ring dataset), both the AD- and ED- extensions can more accurately determine the true labels (TL), whilst the original S, DB, CH result in erroneous assignment (indicated by *) of better scores for other labelsets. Moreover, the AD- extensions offer a clearer separation between the TL and the other labels and offer a significant boost to execution time.

Table 5 – Comparison of original CVIs vs Edging Distance extensions vs Arboris Distance extensions on the ring [97] dataset.

	TL	**FDL**	**SDL**	**HL**	**VL**	**RL**	**Time (s)**
S (↑)	0.268	0.254	0.255	0.254	0.257	0	0.014
ED-S (↑)	0.879	-0.154	-0.194	-0.184	-0.171	-0.209	2.820
AD-S (↑)	0.94	0.003	0.003	0.008	0.013	0.006	0.055
DB (↓)	659.861	1.424*	1.420*	1.427*	1.414*	29.106*	0.001
ED-DB (↓)	0.286	18.018	21.07	16.424	18.393	20.062	1.040
AD-DB (↓)	0.12	17.37	22.453	15.367	17.197	37.897	0.054
CH (↑)	0.002	388.871*	389.559*	386.238*	394.016*	0.815*	0.000
ED-CH (↑)	4098.263	102.526	105.927	130.234	99.984	73.004	1.145
AD-CH (↑)	8309.4	97.364	84.687	105.225	98.268	81.484	0.080

To demonstrate that this is not an exception, the performances of these CVIs on the 32 synthetic benchmark datasets have been aggregated. In Table 6, the number of correct evaluations where the TL obtain the "best" (highest/lowest depending on the CVIs) scores in comparison to the other labelsets and the number of errors where one out of the 5 labelsets obtains a better score than the TL. It is clear that the AD- extensions improve upon the results of the ED- extensions in all cases, and that for both the S and DB they significantly improve the overall performance of the CVIs with a higher number of correct evaluations and a lower number of errors. However, in the case of the CH, neither extension improves the overall results. This decrease for the CH is due to the A1-3 [89], [90] and S1-4 [91] datasets that have a high number of clusters and overlap with regular shapes where the CH fares better (though this is not the case for datasets with irregular shapes).

Table 6 – Aggregated comparison of original CVIs vs ED- extensions vs AD- extensions on the 32 synthetic benchmark datasets.

	S (↑)	ED-S (↑)	AD-S (↑)	DB (↓)	ED-DB (↓)	AD-DB (↓)	CH (↑)	ED-CH (↑)	AD-CH (↑)

Correct evaluations (out of 32)	11	18	**21**	10	13	**18**	9	10	**14**
Errors (out of 205)	51	15	**14**	61	31	**24**	60	49	**43**

We have chosen to exclude the ED- extensions from the rest of the analyses due to the high execution times and the lower performance as shown by the analyses in this subsection.

3.4.1 How dependent is the Arboris Distance on the number of neighbors

One question that easily arises from the proposed approach is how dependent it is on the *n_neighbors* parameter. Due to the use of the MST on the kNN graph, increasing the number of neighbors will induce very few changes as there is only a small probability of better paths to be found. Figure 8 presents an empiric analysis of the parameter for the new CVIs introduced on the simple D1 dataset showing the values obtained for each CVIs by varying the *n_neighbors* parameter in the [3,50] range. The impact is clearly low on all CVIs with no differences in the values obtained for >9, and very small variants for smaller values.

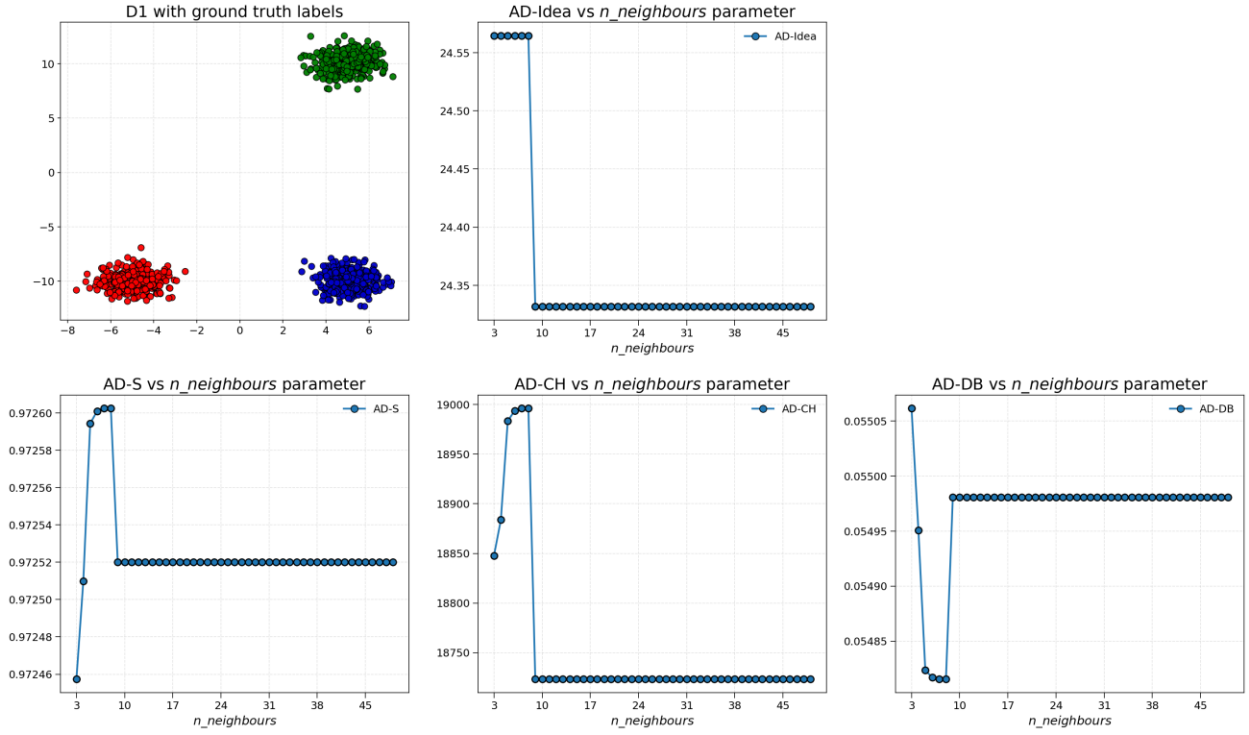

Figure 8 – The impact of *n_neighbors* parameter on the proposed CVIs in the [3,50] range.

3.5 Overall results

The next analysis presented in Table 11 includes 28 CVIs across 32 synthetic benchmark datasets. The performance of each CVIs across the six labelsets was evaluated through the number of correct evaluations where the TL obtain the "best" (highest/lowest depending on the CVIs) scores in comparison to the other labelsets on a dataset and the number of errors where one out of the 5 labelsets obtains a better score than the TL. Our results indicate that the CDbw, DBCV and idea are the highest performing CVIs.

Similarly to the analysis of the DBCV [25], although imperfect [25], [118], [119], we computed Pearson correlations between ARI and each CVI across the 32 datasets. The ARI values were obtained through a grid search of the clustering algorithm for the best scores. The analysis can be separated into the following steps:

- Choose clustering algorithm
- Run grid search to find the best (from the perspective of ARI) parameters for each dataset
- For each CVI:
    - Run clustering algorithm with best parameters on all datasets
    - Compute Pearson correlation between ARI and CVI given the clustering labels

Table 11 – Overall analysis

	Correct evaluations (out of 27)	Errors (out of 135)	Agglomerative	DBSCAN	HDBSCAN	KMeans	MeanShift	Spectral
cSIL (↑)	10	57	-0.297	0.732	0.413	0.262	0.373	0.213
GD43 (↑)	9	65	0.052	0.229	0.217	0.354	0.322	0.197
GD53 (↑)	1	110	0.001	-0.184	0.271	0.007	0.219	0.217
PS (↑)	11	70	-0.059	-0.237	0.072	0.116	-0.072	0.052
rCIP (↓)	5	77	-0.196	0.403	-0.063	0.445	0.134	0.182
WB (↓)	11	56	-0.128	**0.835**	-0.190	0.324	0.195	-0.153
XB (↓)	9	67	-0.183	-0.167	-0.143	0.462	0.344	-0.153
SSE (↓)	11	52	-0.348	0.348	0.351	0.132	-0.090	0.142
RS (↑)	10	60	-0.101	0.601	0.327	0.225	0.391	0.305
DH (↓)	15	35	-0.079	-0.170	0.047	0.269	0.104	0.229
B (↓)	13	47	-0.128	-0.085	-0.135	0.260	0.028	-0.153
BH (↓)	18	21	-0.298	-0.135	-0.084	0.147	-0.191	0.029
D (↑)	12	55	0.093	0.055	0.213	0.274	0.404	0.187
H (↓)	5	71	-0.167	0.517	0.070	0.362	0.480	0.296
DBCV (↑)	25	**2**	0.571	0.437	0.396	0.611	0.537	0.528

I (↑)	10	54	0.102	0.000	0.074	0.329	0.330	0.211
C (↓)	12	52	-0.404	0.063	0.109	0.403	0.263	-0.001
CDbw (↑)	25	3	0.213	0.152	0.196	-0.170	0.027	0.173
VIASCKDE (↑)	16	25	0.180	0.337	0.359	0.228	0.198	0.107
COP (↓)	11	56	-0.262	-0.072	-0.095	-0.112	-0.148	0.085
Sym (↑)	15	32	0.165	0.240	0.087	0.249	0.315	0.303
CS (↓)	12	48	-0.130	-0.109	-0.059	0.257	0.261	-0.151
SF (↑)	5	72	0.196	0.376	0.255	0.409	0.462	0.376
SD (↓)	15	36	-0.143	0.120	0.411	0.378	0.166	0.612
SDbw (↓)	14	38	-0.292	-0.289	-0.238	0.318	0.133	-0.004
XB* (↓)	7	69	-0.184	-0.163	-0.160	0.228	0.479	-0.153
S (↑)	11	52	-0.266	0.350	0.316	0.593	0.474	0.232
DB (↓)	10	62	-0.171	-0.194	-0.188	0.469	0.306	-0.150
CH (↑)	9	61	0.058	0.207	0.194	0.426	0.301	0.125
AD-S (↑)	22	12	0.605	0.449	**0.630**	**0.654**	**0.590**	0.490
AD-DB (↓)	18	24	**0.675**	0.454	0.531	0.475	0.568	**0.569**
AD-CH (↑)	14	45	0.273	0.275	0.180	0.210	0.334	0.178
idea (↑)	**26**	5	0.298	0.201	0.274	-0.022	0.050	0.276

Our evaluation on 32 synthetic datasets indicates that the three CVIs that have the highest number of correct evaluations and the lower number of errors are CDbw, DBCV and idea.

3.6 Real datasets

For the real datasets, Pearson correlations were computed between ARI and each CVI for 6 real datasets across multiple clustering algorithms with multiple parametrizations: DBSCAN, HDBSCAN, MeanShift, AgglomerativeClustering, SpectralClustering, KMeans. The analysis can be separated into the following steps:

- For a given dataset
    - For a given clustering algorithm
        - Run grid search (~500 parametrizations) and save all labelsets
- For each dataset:
    - For each CVI:
        - Compute ARI and CVI given for all the clustering labels regardless of algorithm or parametrization
        - Compute Pearson correlation between ARI and CVI

An important thing to consider when analyzing correlation values is the range of the CVI, for lower-is-better CVIs, negative correlations are relevant otherwise for positive correlations it means that the metric cannot identify correctly the performance of the clustering. The analogous situation for higher-is-better CVIs, negative correlations indicate incorrect evaluation.

For lower-is-better CVIs, all correlations have been negated.

CDbw is unable to run on high dimensional datasets (modifications can make it run but takes fucking long)

Table 12 -

	ecoli	**glass**	**yeast**	**statlog**	**wdbc**	**wine**	**ionosphere**	**sonar**
cSIL	-0.177	-0.057	-0.890	0.241	-0.429	-0.229	0.015	-0.578
GD43	-0.150	-0.152	0.153	-0.212	-0.782	0.529	-0.515	-0.518
GD53	-0.550	-0.441	0.072	0.377	-0.410	0.428	-0.400	-0.039
PS	-0.446	-0.205	0.460	-0.018	-0.107	0.020	-0.533	0.155
rCIP	0.128	-0.047	-0.276	-0.057	-0.681	-0.145	-0.396	0.020
WB	0.686	0.692	0.681	0.707	0.694	0.444	0.132	0.247
XB	0.090	0.072	0.049	0.100	-0.730	0.442	-0.469	-0.383
SSE	0.682	-0.571	0.598	0.587	0.457	0.421	-0.445	0.248
RS	-0.286	-0.375	-0.410	0.060	-0.664	0.304	-0.620	-0.160
DH	-0.343	-0.523	0.004	-0.189	-0.070	0.094	-0.552	0.305
B	-0.048	-0.507	0.216	0.184	0.527	0.250	-0.720	0.126
BH	0.743	0.068	0.033	-0.160	0.441	0.503	0.100	0.612
D	-0.651	0.004	-0.499	-0.270	-0.352	-0.127	-0.107	-0.124
H	-0.200	-0.425	-0.394	-0.060	-0.816	-0.028	-0.634	-0.180
DBCV	-0.701	0.402	-0.690	-0.706	-0.822	-0.091	0.199	-0.020
I	-0.112	-0.135	-0.392	-0.201	-0.034	-0.098	-0.179	-0.079
C	-0.165	**0.735**	0.781	0.370	0.212	0.444	0.393	-0.570
CDbw								
VIASCKDE	0.503	0.294	0.592	0.094	-0.109	0.250	-0.514	-0.471
COP	0.511	-0.348	0.517	0.681	-0.055	**0.638**	-0.170	0.745
Sym	-0.342	0.330	0.608	0.253	-0.123	0.097	0.214	-0.638
CS	-0.390	-0.347	0.119	-0.061	0.052	0.022	-0.649	0.051
SF	-0.779	-0.030	-0.067	0.208	-0.737	-0.372	-0.183	-0.184
SD	-0.005	-0.438	0.262	0.085	0.255	0.290	-0.550	0.495
SDbw	-0.211	0.004	-0.144	-0.161	0.360	0.298	-0.512	-1.000
XB*	-0.225	0.431	0.354	0.277	0.075	0.331	0.243	-0.729
S	-0.351	0.693	0.718	0.238	-0.197	0.175	0.439	-0.528
DB	-0.390	-0.413	0.257	0.176	0.057	0.209	-0.595	-0.121
CH	**0.748**	0.697	**0.837**	**0.825**	**0.790**	0.254	0.178	-0.011
AD-S	-0.758	0.460	-0.366	-0.442	-0.566	-0.302	0.080	-0.382
AD-DB	-0.590	-0.345	-0.516	-0.343	-0.434	-0.034	0.055	-0.326
AD-CH	0.350	0.699	0.156	0.676	-0.152	-0.157	**0.532**	-0.355
AD-idea	-0.209	-0.485	-0.271	-0.035	-0.067	-0.294	-0.308	**0.639**

4 Conclusions

CVIs are sensitive to the number of clusters, perhaps too sensitive to use them as validation for the choice of this parameter. Internal CVIs tend to be sensitive to cluster overlap resulting in erroneous performance evaluations, while external CVIs tend to be sensitive to cluster imbalance resulting in erroneous performance evaluations.

5 Bibliography

[1] G. S. Linoff and Michael J. A. Berry, *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*, 3rd ed. Wiley Publishing, 2011.

[2] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of Internal Clustering Validation Measures," in *2010 IEEE International Conference on Data Mining*, Dec. 2010, pp. 911–916. doi: 10.1109/ICDM.2010.35.

[3] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On Clustering Validation Techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2, pp. 107–145, Dec. 2001, doi: 10.1023/A:1012801612483.

[4] R. Todeschini, D. Ballabio, V. Termopoli, and V. Consonni, "Extended multivariate comparison of 68 cluster validity indices. A review," *Chemom. Intell. Lab. Syst.*, vol. 251, p. 105117, Aug. 2024, doi: 10.1016/j.chemolab.2024.105117.

[5] M. Rezaei and P. Fränti, "Set Matching Measures for External Cluster Validity," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2173–2186, Aug. 2016, doi: 10.1109/TKDE.2016.2551240.

[6] S. Theodoridis and K. Koutroumbas, "Chapter 16 - Cluster Validity," in *Pattern Recognition (Fourth Edition)*, S. Theodoridis and K. Koutroumbas, Eds., Boston: Academic Press, 2009, pp. 863–913. doi: 10.1016/B978-1-59749-272-0.50018-9.

[7] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," *SIGMOD Rec*, vol. 27, no. 2, pp. 73–84, June 1998, doi: 10.1145/276305.276312.

[8] M. Ramze Rezaee, B. P. F. Lelieveldt, and J. H. C. Reiber, "A new cluster validity index for the fuzzy *c*-mean," *Pattern Recognit. Lett.*, vol. 19, no. 3, pp. 237–246, Mar. 1998, doi: 10.1016/S0167-8655(97)00168-2.

[9] S. Dasgupta, "The hardness of k-means clustering".

[10] A. Deshpande, A. Louis, and A. V. Singh, "On Euclidean k-Means Clustering with α-Center Proximity," arXiv.org. Accessed: Nov. 13, 2025. [Online]. Available: https://arxiv.org/abs/1804.10827v3

[11] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k<math><mi is="true">k</mi></math>-means problem is NP-hard," *Theor. Comput. Sci.*, vol. 442, pp. 13–21, July 2012, doi: 10.1016/j.tcs.2010.05.034.

[12] A. José-García and W. Gómez-Flores, "A survey of cluster validity indices for automatic data clustering using differential evolution," in *Proceedings of the Genetic and Evolutionary Computation Conference*, in GECCO '21. New York, NY, USA: Association for Computing Machinery, June 2021, pp. 314–322. doi: 10.1145/3449639.3459341.

[13] E. Falkenauer, *Genetic Algorithms and Grouping Problems*. USA: John Wiley & Sons, Inc., 1998.
[14] M. Gagolewski, M. Bartoszuk, and A. Cena, "Are cluster validity measures (in) valid?," *Inf. Sci.*, vol. 581, pp. 620–636, Dec. 2021, doi: 10.1016/j.ins.2021.10.004.
[15] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Mach. Learn.*, vol. 75, no. 2, pp. 245–248, May 2009, doi: 10.1007/s10994-009-5103-0.
[16] M. Garey, D. Johnson, and H. Witsenhausen, "The complexity of the generalized Lloyd - Max problem (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 255–256, Mar. 1982, doi: 10.1109/TIT.1982.1056488.
[17] A. M. Ikotun, F. Habyarimana, and A. E. Ezugwu, "Benchmarking validity indices for evolutionary K-means clustering performance," *Sci. Rep.*, vol. 15, no. 1, p. 21842, July 2025, doi: 10.1038/s41598-025-08473-6.
[18] B. A. Hassan, N. B. Tayfor, A. A. Hassan, A. M. Ahmed, T. A. Rashid, and N. N. Abdalla, "From A-to-Z review of clustering validation indices," *Neurocomputing*, vol. 601, p. 128198, Oct. 2024, doi: 10.1016/j.neucom.2024.128198.
[19] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, and S. Wu, "Understanding and Enhancement of Internal Clustering Validation Measures," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 982–994, June 2013, doi: 10.1109/TSMCB.2012.2220543.
[20] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, Dec. 2002, doi: 10.1109/TPAMI.2002.1114856.
[21] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, June 1985, doi: 10.1007/BF02294245.
[22] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognit.*, vol. 46, no. 1, pp. 243–256, Jan. 2013, doi: 10.1016/j.patcog.2012.07.021.
[23] A. M. Ikotun, F. Habyarimana, and A. E. Ezugwu, "Cluster validity indices for automatic clustering: A comprehensive review," *Heliyon*, vol. 11, no. 2, p. e41953, Jan. 2025, doi: 10.1016/j.heliyon.2025.e41953.
[24] R. Xu, J. Xu, and D. C. Wunsch, "A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 42, no. 4, pp. 1243–1256, Aug. 2012, doi: 10.1109/TSMCB.2012.2188509.
[25] D. Moulavi, P. A. Jaskowiak, R. J. G. B. Campello, A. Zimek, and J. Sander, "Density-Based Clustering Validation," in *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)*, in Proceedings. , Society for Industrial and Applied Mathematics, 2014, pp. 839–847. doi: 10.1137/1.9781611973440.96.
[26] S.R.Pande, M. S.S.Sambare, and V.M.Thakre, "Data Clustering Using Data Mining Techniques," 2012. Accessed: Dec. 14, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Data-Clustering-Using-Data-Mining-Techniques-S.R.Pande-S.S.Sambare/ed3135bcaf33fce1f4be228e6f28667dd6e4f2ac
[27] S. Liang, D. Han, and Y. Yang, "Cluster validity index for irregular clustering results," *Appl. Soft Comput.*, vol. 95, p. 106583, Oct. 2020, doi: 10.1016/j.asoc.2020.106583.

[28] C. Fang, L. Zhou, X. Gu, X. Liu, and M. Werner, "A data driven approach to urban area delineation using multi source geospatial data," *Sci. Rep.*, vol. 15, no. 1, p. 8708, Mar. 2025, doi: 10.1038/s41598-025-93366-x.

[29] M. van Mulken, J. Eikelboom, K. Verbeek, B. Speckmann, and F. Van Langevelde, "Quantifying the spatial scales of animal clusters using density surfaces," *J. R. Soc. Interface*, vol. 22, no. 230, p. 20250274, Sept. 2025, doi: 10.1098/rsif.2025.0274.

[30] A. Elvitigala, U. dani Navaratne, S. Rathnayake, and K. Dissanayaka, "Galaxy Clustering and Classification using Machine Learning Algorithms and XAI," in *2024 9th International Conference on Information Technology Research (ICITR)*, Dec. 2024, pp. 1–6. doi: 10.1109/ICITR64794.2024.10857763.

[31] V. Singhal *et al.*, "BANKSY unifies cell typing and tissue domain segmentation for scalable spatial omics data analysis," *Nat. Genet.*, vol. 56, no. 3, pp. 431–441, Mar. 2024, doi: 10.1038/s41588-024-01664-3.

[32] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, June 2002, doi: 10.1073/pnas.122653799.

[33] Z. Botta-Dukát, "A new approach for evaluating internal cluster validation indices," Aug. 02, 2023, *arXiv*: arXiv:2308.03894. doi: 10.48550/arXiv.2308.03894.

[34] P. A. Jaskowiak, D. Moulavi, A. C. S. Furtado, R. J. G. B. Campello, A. Zimek, and J. Sander, "On strategies for building effective ensembles of relative clustering validity criteria," *Knowl. Inf. Syst.*, vol. 47, no. 2, pp. 329–354, May 2016, doi: 10.1007/s10115-015-0851-6.

[35] H. Jeon, M. Aupetit, D. Shin, A. Cho, S. Park, and J. Seo, "Sanity Check for External Clustering Validation Benchmarks using Internal Validation Measures," Sept. 20, 2022, *arXiv*: arXiv:2209.10042. doi: 10.48550/arXiv.2209.10042.

[36] P. A. Jaskowiak, I. G. Costa, and R. J. G. B. Campello, "The area under the ROC curve as a measure of clustering quality," *Data Min. Knowl. Discov.*, vol. 36, no. 3, pp. 1219–1245, May 2022, doi: 10.1007/s10618-022-00829-0.

[37] Y. Lei, J. C. Bezdek, S. Romano, N. X. Vinh, J. Chan, and J. Bailey, "Ground truth bias in external cluster validity indices," *Pattern Recognit.*, vol. 65, pp. 58–70, May 2017, doi: 10.1016/j.patcog.2016.12.003.

[38] M. C. P. de Souto, A. L. V. Coelho, K. Faceli, T. C. Sakata, V. Bonadia, and I. G. Costa, "A Comparison of External Clustering Evaluation Indices in the Context of Imbalanced Data Sets," in *2012 Brazilian Symposium on Neural Networks*, Oct. 2012, pp. 49–54. doi: 10.1109/SBRN.2012.25.

[39] M. Gagolewski, "Normalised Clustering Accuracy: An Asymmetric External Cluster Validity Measure," *J. Classif.*, vol. 42, no. 1, pp. 2–30, Mar. 2025, doi: 10.1007/s00357-024-09482-2.

[40] E.-R. Ardelean, R. L. Portase, R. Potolea, and M. Dînșoreanu, "A path-based distance computation for non-convexity with applications in clustering," *Knowl. Inf. Syst.*, vol. 67, no. 2, pp. 1415–1453, Feb. 2025, doi: 10.1007/s10115-024-02275-4.

[41] C.-H. Chou, M.-C. Su, and E. Lai, "A new cluster validity measure and its application to image compression," *Pattern Anal. Appl.*, vol. 7, no. 2, pp. 205–220, July 2004, doi: 10.1007/s10044-004-0218-1.

[42] M. Halkidi and M. Vazirgiannis, “A density-based cluster validity approach using multi-representatives,” *Pattern Recognit. Lett.*, vol. 29, no. 6, pp. 773–786, Apr. 2008, doi: 10.1016/j.patrec.2007.12.011.

[43] K. R. Žalik and B. Žalik, “Validity index for clusters of different sizes and densities,” *Pattern Recognit. Lett.*, vol. 32, no. 2, pp. 221–234, Jan. 2011, doi: 10.1016/j.patrec.2010.08.007.

[44] N. R. Pal and J. Biswas, “Cluster validation using graph theoretic concepts,” *Pattern Recognit.*, vol. 30, no. 6, pp. 847–857, June 1997, doi: 10.1016/S0031-3203(96)00127-6.

[45] E. J. Pauwels and G. Frederix, “Finding Salient Regions in Images: Nonparametric Clustering for Image Segmentation and Grouping,” *Comput. Vis. Image Underst.*, vol. 75, no. 1, pp. 73–85, July 1999, doi: 10.1006/cviu.1999.0763.

[46] 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, no. 1, p. 4059302, 2022, doi: 10.1155/2022/4059302.

[47] J. Yang and I. Lee, “Cluster validity through graph-based boundary analysis: Proceedings of the International Conference on Information and Knowledge Engineering, IKE’04,” *Proc. Int. Conf. Inf. Knowl. Eng. IKE04*, pp. 204–210, 2004.

[48] E. Dimitriadou, S. Dolničar, and A. Weingessel, “An examination of indexes for determining the number of clusters in binary data sets,” *Psychometrika*, vol. 67, no. 1, pp. 137–159, Mar. 2002, doi: 10.1007/BF02294713.

[49] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5.1, University of California Press, 1967, pp. 281–298. Accessed: Sept. 03, 2025. [Online]. Available: https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992

[50] T. Ullmann, A. Beer, M. Hünemörder, T. Seidl, and A.-L. Boulesteix, “Over-optimistic evaluation and reporting of novel cluster algorithms: an illustrative study,” *Adv. Data Anal. Classif.*, vol. 17, no. 1, pp. 211–238, Mar. 2023, doi: 10.1007/s11634-022-00496-5.

[51] M. Brun *et al.*, “Model-based evaluation of clustering validation measures,” *Pattern Recognit.*, vol. 40, no. 3, pp. 807–824, Mar. 2007, doi: 10.1016/j.patcog.2006.06.026.

[52] M. Kim and R. S. Ramakrishna, “New indices for cluster validity assessment,” *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2353–2363, Nov. 2005, doi: 10.1016/j.patrec.2005.04.007.

[53] H. Li, S. Zhang, X. Ding, C. Zhang, and P. Dale, “Performance Evaluation of Cluster Validity Indices (CVIs) on Multi/Hyperspectral Remote Sensing Datasets,” *Remote Sens.*, vol. 8, no. 4, p. 295, Apr. 2016, doi: 10.3390/rs8040295.

[54] Z. Ansari, M. F. Azeem, W. Ahmed, and A. V. Babu, “Quantitative Evaluation of Performance and Validity Indices for Clustering the Web Navigational Sessions,” July 13, 2015, *arXiv*: arXiv:1507.03340. doi: 10.48550/arXiv.1507.03340.

[55] U. Maulik and S. Bandyopadhyay, “Performance evaluation of some clustering algorithms and validity indices,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, Dec. 2002, doi: 10.1109/TPAMI.2002.1114856.

[56] Z. Wang and C. Ye, “Deep Clustering Evaluation: How to Validate Internal Clustering Validation Measures,” Mar. 21, 2024, *arXiv*: arXiv:2403.14830. doi: 10.48550/arXiv.2403.14830.
[57] T. Caliński and H. JA, “A Dendrite Method for Cluster Analysis,” *Commun. Stat. - Theory Methods*, vol. 3, pp. 1–27, Jan. 1974, doi: 10.1080/03610927408827101.
[58] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979, doi: 10.1109/TPAMI.1979.4766909.
[59] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.
[60] A. Rosenberg and J. Hirschberg, “V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure.,” Jan. 2007, pp. 410–420.
[61] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, “Internal versus External cluster validation indexes,” vol. 5, no. 1, p. 8, 2011.
[62] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.
[63] G. W. Milligan and M. C. Cooper, “An examination of procedures for determining the number of clusters in a data set,” *Psychometrika*, vol. 50, no. 2, pp. 159–179, June 1985, doi: 10.1007/BF02294245.
[64] G. Ball and D. J. Hall, “ISODATA, A NOVEL METHOD OF DATA ANALYSIS AND PATTERN CLASSIFICATION,” Apr. 1965. Accessed: Nov. 04, 2025. [Online]. Available: https://www.semanticscholar.org/paper/ISODATA%2C-A-NOVEL-METHOD-OF-DATA-ANALYSIS-AND-Ball-Hall/7dac28166b91d89ef6c38cf4fbb9f647b1d73c61
[65] N. V. Thieu, “PerMetrics: A Framework of Performance Metrics for Machine Learning Models,” *J. Open Source Softw.*, vol. 9, no. 95, p. 6143, Mar. 2024, doi: 10.21105/joss.06143.
[66] X. L. Xie and G. Beni, “A validity measure for fuzzy clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 841–847, Aug. 1991, doi: 10.1109/34.85677.
[67] *cvi: A Python package for both batch and incremental cluster validity indices.* Python. Accessed: Nov. 04, 2025. [Online]. Available: https://github.com/AP6YC/cvi
[68] N. Galmiche, “PyCVI: A Python package for internal Cluster Validity Indices, compatible with time-series data,” *J. Open Source Softw.*, vol. 9, no. 102, p. 6841, Oct. 2024, doi: 10.21105/joss.06841.
[69] J. C. Dunn, “A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters,” *J. Cybern.*, vol. 3, no. 3, pp. 32–57, Jan. 1973, doi: 10.1080/01969727308546046.
[70] J. C. Dunn†, “Well-Separated Clusters and Optimal Fuzzy Partitions,” *J. Cybern.*, vol. 4, no. 1, pp. 95–104, Jan. 1974, doi: 10.1080/01969727408546059.
[71] Q. Zhao, M. Xu, and P. Fränti, “Sum-of-Squares Based Cluster Validity Index and Significance Analysis,” in *Adaptive and Natural Computing Algorithms*, M. Kolehmainen, P. Toivanen, and B. Beliczynski, Eds., Berlin, Heidelberg: Springer, 2009, pp. 313–322. doi: 10.1007/978-3-642-04921-7_32.
[72] M. Thompson, R. O. Duda, and P. E. Hart, “Pattern Classification and Scene Analysis,” in *Leonardo*, 1974, p. 370. doi: 10.2307/1573081.
[73] E. M. L. Beale, *Euclidean Cluster Analysis*. Scientific Control Systems Limited, 1969.

[74] J. A. Hartigan, *Clustering Algorithms*, 99th ed. USA: John Wiley & Sons, Inc., 1975.
[75] M. Rawashdeh and A. Ralescu, "Center-Wise Intra-Inter Silhouettes," in *Scalable Uncertainty Management*, E. Hüllermeier, S. Link, T. Fober, and B. Seeger, Eds., Berlin, Heidelberg: Springer, 2012, pp. 406–419. doi: 10.1007/978-3-642-33362-0_31.
[76] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 28, no. 3, pp. 301–315, June 1998, doi: 10.1109/3477.678624.
[77] M.-S. Yang and K.-L. Wu, "A new validity index for fuzzy clustering," in *10th IEEE International Conference on Fuzzy Systems. (Cat. No.01CH37297)*, Dec. 2001, pp. 89–92 vol.1. doi: 10.1109/FUZZ.2001.1007254.
[78] M. Moshtaghi, J. C. Bezdek, S. M. Erfani, C. Leckie, and J. Bailey, "Online Cluster Validity Indices for Streaming Data," Jan. 08, 2018, *arXiv*: arXiv:1801.02937. doi: 10.48550/arXiv.1801.02937.
[79] Q. Zhao, M. Xu, and P. Fränti, "Sum-of-Squares Based Cluster Validity Index and Significance Analysis," in *Adaptive and Natural Computing Algorithms*, M. Kolehmainen, P. Toivanen, and B. Beliczynski, Eds., Berlin, Heidelberg: Springer, 2009, pp. 313–322. doi: 10.1007/978-3-642-04921-7_32.
[80] S. Saitta, B. Raphael, and I. F. C. Smith, "A Bounded Index for Cluster Validity," in *Machine Learning and Data Mining in Pattern Recognition*, P. Perner, Ed., Berlin, Heidelberg: Springer, 2007, pp. 174–187. doi: 10.1007/978-3-540-73499-4_14.
[81] M. Halkidi, M. Vazirgiannis, and Y. Batistakis, "Quality Scheme Assessment in the Clustering Process," in *Principles of Data Mining and Knowledge Discovery*, D. A. Zighed, J. Komorowski, and J. Żytkow, Eds., Berlin, Heidelberg: Springer, 2000, pp. 265–276. doi: 10.1007/3-540-45372-5_26.
[82] M. Halkidi and M. Vazirgiannis, "Clustering validity assessment: finding the optimal partitioning of a data set," in *Proceedings 2001 IEEE International Conference on Data Mining*, Nov. 2001, pp. 187–194. doi: 10.1109/ICDM.2001.989517.
[83] L. J. Hubert and J. R. Levin, "A general statistical framework for assessing categorical clustering in free recall," *Psychol. Bull.*, vol. 83, no. 6, pp. 1072–1080, 1976, doi: 10.1037/0033-2909.83.6.1072.
[84] I. Gurrutxaga *et al.*, "SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index," *Pattern Recognit.*, vol. 43, no. 10, pp. 3364–3373, Oct. 2010, doi: 10.1016/j.patcog.2010.04.021.
[85] D. Chicco, G. Sabino, L. Oneto, and G. Jurman, "The DBCV index is more informative than DCSI, CDbw, and VIASCKDE indices for unsupervised clustering internal assessment of concave-shaped and density-based clusters," *PeerJ Comput. Sci.*, vol. 11, p. e3095, Aug. 2025, doi: 10.7717/peerj-cs.3095.
[86] J. Gauss, F. Scheipl, and M. Herrmann, "DCSI -- An improved measure of cluster separability based on separation and connectedness," Apr. 11, 2025, *arXiv*: arXiv:2310.12806. doi: 10.48550/arXiv.2310.12806.
[87] S. Chowdhury and R. C. de Amorim, "An efficient density-based clustering algorithm using reverse nearest neighbour," Nov. 19, 2018, *arXiv*: arXiv:1811.07615. doi: 10.48550/arXiv.1811.07615.
[88] H. Movahedi-Lankarani, "Ultrametric spaces and the logarithmic ratio," Dec. 18, 2025, *arXiv*: arXiv:2512.16820. doi: 10.48550/arXiv.2512.16820.
[89] I. Kärkkäinen and P. Fränti, "A Dynamic local search algorithm for the clustering problem," 2002. Accessed: Nov. 13, 2025. [Online]. Available:

https://www.semanticscholar.org/paper/A-Dynamic-local-search-algorithm-for-the-clustering-K%C3%A4rkk%C3%A4inen-Fr%C3%A4nti/4c236eaf6a1db6fec5f032f0bceb3ffd54080aee
[90] M. Gagolewski, "A framework for benchmarking clustering algorithms," *SoftwareX*, vol. 20, p. 101270, Dec. 2022, doi: 10.1016/j.softx.2022.101270.
[91] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognit.*, vol. 39, no. 5, pp. 761–775, May 2006, doi: 10.1016/j.patcog.2005.09.012.
[92] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans Knowl Discov Data*, vol. 1, no. 1, pp. 4-es, Mar. 2007, doi: 10.1145/1217299.1217303.
[93] C. T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. Comput.*, vol. C–20, no. 1, pp. 68–86, Jan. 1971, doi: 10.1109/T-C.1971.223083.
[94] C. J. Veenman, M. J. T. Reinders, and E. Backer, "A maximum variance cluster algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1273–1280, Sept. 2002, doi: 10.1109/TPAMI.2002.1033218.
[95] A. K. Jain and M. H. C. Law, "Data Clustering: A User's Dilemma," in *Pattern Recognition and Machine Intelligence*, S. K. Pal, S. Bandyopadhyay, and S. Biswas, Eds., Berlin, Heidelberg: Springer, 2005, pp. 1–10. doi: 10.1007/11590316_1.
[96] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognit.*, vol. 41, no. 1, pp. 191–203, Jan. 2008, doi: 10.1016/j.patcog.2007.04.010.
[97] D. Graves and W. Pedrycz, "Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study," *Fuzzy Sets Syst.*, vol. 161, no. 4, pp. 522–543, Feb. 2010, doi: 10.1016/j.fss.2009.10.021.
[98] L. Fu and E. Medico, "FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinformatics*, vol. 8, no. 1, p. 3, Jan. 2007, doi: 10.1186/1471-2105-8-3.
[99] A. Ultsch, "Clustering wih som: U* c," in *Proceedings of the 5th workshop on self-organizing maps*, 2005, pp. 75–82. Accessed: Jan. 02, 2026. [Online]. Available: https://www.researchgate.net/profile/Alfred-Ultsch/publication/229048370_CLUSTERING_WIH_SOM_U_C/links/58da1d4ca6fdccca1c4c0a4d/CLUSTERING-WIH-SOM-U-C.pdf
[100] N. Macià and E. Bernadó-Mansilla, "Towards UCI+: A mindful repository design," *Inf. Sci.*, vol. 261, pp. 237–262, Mar. 2014, doi: 10.1016/j.ins.2013.08.059.
[101] S. Chang, Y. Shihong, and L. Qi, "Clustering Characteristics of UCI Dataset," in *2020 39th Chinese Control Conference (CCC)*, July 2020, pp. 6301–6306. doi: 10.23919/CCC50068.2020.9189507.
[102] K. Nakai, "Ecoli." UCI Machine Learning Repository, 1996. doi: 10.24432/C5388M.
[103] B. German, "Glass Identification." UCI Machine Learning Repository, 1987. doi: 10.24432/C5WW2P.
[104] K. Nakai, "Yeast." UCI Machine Learning Repository, 1991. doi: 10.24432/C5KG68.
[105] Unknown, "Statlog (Image Segmentation)." UCI Machine Learning Repository, 1990. doi: 10.24432/C5P01G.
[106] O. M. William Wolberg, "Breast Cancer Wisconsin (Diagnostic)." UCI Machine Learning Repository, 1993. doi: 10.24432/C5DW2B.
[107] M. F. Stefan Aeberhard, "Wine." UCI Machine Learning Repository, 1992. doi: 10.24432/C5PC7J.

[108] R. G. Terry Sejnowski, “Connectionist Bench (Sonar, Mines vs. Rocks).” UCI Machine Learning Repository, 1988. doi: 10.24432/C5T01Q.
[109] S. W. V. Sigillito, “Ionosphere.” UCI Machine Learning Repository, 1989. doi: 10.24432/C5W01B.
[110] E. B. Fowlkes and C. L. Mallows, “A Method for Comparing Two Hierarchical Clusterings,” *J. Am. Stat. Assoc.*, vol. 78, no. 383, pp. 553–569, 1983, doi: 10.2307/2288117.
[111] D. Steinley, “Properties of the Hubert-Arable Adjusted Rand Index,” *Psychol. Methods*, vol. 9, no. 3, pp. 386–396, 2004, doi: 10.1037/1082-989X.9.3.386.
[112] L. Hubert and P. Arabie, “Comparing partitions,” *J. Classif.*, vol. 2, no. 1, pp. 193–218, Dec. 1985, doi: 10.1007/BF01908075.
[113] N. X. Vinh, J. Epps, and J. Bailey, “Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance,” p. 18.
[114] A. Strehl and J. Ghosh, “Cluster Ensembles --- A Knowledge Reuse Framework for Combining Multiple Partitions,” *J. Mach. Learn. Res.*, vol. 3, no. Dec, pp. 583–617, 2002.
[115] N. Vinh, J. Epps, and J. Bailey, *Information theoretic measures for clusterings comparison: Is a correction for chance necessary?* 2009. doi: 10.1145/1553374.1553511.
[116] D. Lazarenko and T. Bonald, *Pairwise Adjusted Mutual Information*. 2021.
[117] P. Fränti and S. Sieranoja, “K-means properties on six clustering benchmark datasets,” *Appl. Intell.*, vol. 48, no. 12, pp. 4743–4759, Dec. 2018, doi: 10.1007/s10489-018-1238-7.
[118] D. Moulavi, P. A. Jaskowiak, R. J. G. B. Campello, A. Zimek, and J. Sander, “Density-Based Clustering Validation,” in *Proceedings of the 2014 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, Apr. 2014, pp. 839–847. doi: 10.1137/1.9781611973440.96.
[119] I. Färber *et al.*, “On using class-labels in evaluation of clusterings,” Jan. 2010.
[120] Kelly, Markelle, Longjohn, Rachel, and Nottingham Kolby, “Home - UCI Machine Learning Repository.” Accessed: Mar. 23, 2024. [Online]. Available: https://archive.ics.uci.edu/
[121] A. B. Said, R. Hadjidj, and S. Foufou, “Cluster validity index based on Jeffrey divergence,” *Pattern Anal. Appl.*, vol. 20, no. 1, pp. 21–31, Feb. 2017, doi: 10.1007/s10044-015-0453-7.
[122] M. M. ElMorshedy, R. Fathalla, and Y. El-Sonbaty, “Feature Transformation Framework for Enhancing Compactness and Separability of Data Points in Feature Space for Small Datasets,” *Appl. Sci.*, vol. 12, no. 3, p. 1713, Jan. 2022, doi: 10.3390/app12031713.
[123] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, in KDD’96. Portland, Oregon: AAAI Press, Aug. 1996, pp. 226–231.
[124] F. Boudane and A. Berrichi, “Gabriel graph-based connectivity and density for internal validity of clustering,” *Prog. Artif. Intell.*, vol. 9, no. 3, pp. 221–238, Sept. 2020, doi: 10.1007/s13748-020-00209-z.

Data Availability

The datasets used in this work are openly available and can be found at:

- Synthetic datasets [90]: https://github.com/gagolews/clustering-data-v1/
- Real datasets [120]: https://archive.ics.uci.edu/datasets

Code Availability

The code that supports the findings of this work was written in Python and is openly available at: https://github.com/ArdeleanRichard/Clustering-Validity/.

Competing Interests

The authors have declared that no competing interests exist.

Authors' contributions

Conceptualization, E.-R.A.; methodology, E.-R.A.; software, E.-R.A. and R.L.P.; validation, E.-R.A. and R.L.P.; formal analysis, E.-R.A., M.S. and R.L.P.; investigation, E.-R.A. and R.L.P.; data curation, E.-R.A. and R.L.P.; writing—original draft preparation, E.-R.A. and R.L.P.; writing—review and editing, E.-R.A., M.S. and R.L.P.; visualization, E.-R.A. and R.L.P.; supervision, E.-R.A. and R.L.P.; project administration, E.-R.A. and R.L.P. All authors have read and agreed to the published version of the manuscript.

Funding

This work is supported by the project "Romanian Hub for Artificial Intelligence-HRIA", Smart Growth, Digitization and Financial Instruments Program, MySMIS no. 334906.

Supplementary

Other applications of AD

The AD can be used to substitute distance computation, another use case is the K-Means [49] clustering algorithm as it is known to have issues with irregular shaped clusters and imbalanced clusters [117].

The analysis compares the comparison of K-Means, DBSCAN, HDBSCAN, MeanShift, AgglomerativeClustering, SpectralClustering and the ED- and AD- extensions of K-Means from the perspective of ARI (Table S1), AMI (Table S2) and execution time (Table S3) on six datasets: unbalance [5], aggregation [92], compound [93], jain [95], pathbased [96], spiral [96].

Parameter values were found with grid search for DBSCAN, HDBSCAN, MeanShift, AgglomerativeClustering, SpectralClustering due to their high number of and complex parameters. AD-K-Means has a single parameter *n_neighbours* (excluding the *k* of K-Means) which as shown above does not affect significantly the results, making it much easier to use.

The scores obtained by ARI (Table S1) and AMI (Table S2) for these clustering algorithms indicate that the AD-extension of K-Means outperforms K-Means on all datasets and SpectralClustering and MeanShift on almost all datasets. However, DBSCAN, HDBSCAN and AgglomerativeClustering obtain better scores on most cases.

Table S1 – Comparison of clustering algorithms by AMI.

algorithm	**aggregation**	**compound**	**jain**	**pathbased**	**spiral**	**unbalance**
K-Means	0.838	0.725	0.526	0.543	-0.005	**1**
DBSCAN	**0.983**	**0.908**	**1**	**0.887**	**1**	0.999
HDBSCAN	0.898	0.839	0.934	0.725	**1**	**1**
MeanShift	0.889	0.796	0.433	0.578	0.236	**1**
AgglomerativeClustering	0.928	0.813	0.928	0.563	**1**	0.974
SpectralClustering	0.774	0.862	0.667	0.713	**1**	0.796
ED-K-Means	0.818	0.746	**1**	0.524	0.404	0.923
AD-K-Means	0.873	0.85	0.832	0.663	**1**	**1**

Table S2 – Comparison of clustering algorithms by ARI.

algorithm	**aggregation**	**compound**	**jain**	**pathbased**	**spiral**	**unbalance**
K-Means	0.735	0.583	0.577	0.461	-0.006	**1**
DBSCAN	**0.987**	**0.949**	**1**	**0.92**	**1**	**1**
HDBSCAN	0.827	0.883	0.976	0.682	**1**	**1**
MeanShift	0.868	0.77	0.367	0.574	0.113	**1**
AgglomerativeClustering	0.929	0.796	0.976	0.485	**1**	0.995
SpectralClustering	0.539	0.807	0.733	0.657	**1**	0.61
ED-K-Means	0.652	0.567	1	0.449	0.353	0.886
AD-K-Means	0.827	0.831	0.899	0.58	**1**	**1**

Table S3 – Comparison of clustering algorithms by time in seconds.

algorithm	**aggregation**	**compound**	**jain**	**pathbased**	**spiral**	**unbalance**
K-Means	0.047	0.346	0.051	0.019	0.245	**0.034**
DBSCAN	0.008	0.003	0.004	0.003	0.003	0.343

HDBSCAN	0.01	0.006	0.006	0.007	0.008	0.125
MeanShift	0.178	0.147	0.127	0.086	0.098	0.25
AgglomerativeClustering	0.019	0.007	0.002	0.001	0.001	0.612
SpectralClustering	0.126	0.062	0.091	0.027	0.027	1.013
ED-K-Means	96.392	23.45	3.839	2.055	5.959	3820.879
AD-K-Means	0.923	0.397	0.046	0.052	0.066	87.619

Mathematical formulations of CVIs

The following equations present the calculation of means for the whole dataset and for a cluster itself also known as the cluster mean, cluster center or centroid:

$$\mu_{data} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\mu_k = \frac{1}{n_k}\sum_{x_k \in C_k} x_k$$

As mentioned above, internal CVIs are based upon the concepts of intra-cluster distances (i.e. cohesiveness / compactness) and inter-cluster distances (i.e. separation).

Compactness measures the degree of closeness between the data points of clusters through the sum of squared errors (SSE), also known as the sum of squares within or within-cluster sum of squares (SSW) [121]. A clustering partition is considered to be good when its variance or SSE is low; thus, compactness is a minimization measure, with perfect scores obtained when all the points of a cluster are duplicated (or there is a single point). It is computed as the total sum of squared distances between a data point of a given cluster and the centroid of said cluster, as given by the following equations:

$$SSE(C_k) = \sum_{x \epsilon C_k} d(x, \mu_k)^2$$

$$SSE = SSW = \sum_{k=1}^{K} SSE(C_k) = \sum_{k=1}^{K}\sum_{x \epsilon C_k} d(x, \mu_k)^2$$

Separation measures the distance between clusters using the sum of squares between (SSB), also known as between-cluster sum of squares (BSS) [122]. A clustering partition is considered to be good when SSB values are high; thus, separation is a maximization measure. It is typically

computed as the sum of squares distances between cluster centroids and the global mean of the dataset, as given by the following equation:

$$SSB = BSS = \sum_{k=1}^{K} n_k * d(\mu_k, \mu_{data})^2$$

Davies-Bouldin index (DB) [3], [57], [58] is computed as the average similarity of clusters. The similarity is computed using the distance between clusters and their sizes. DBS has an inverse performance interval to the other metrics presented in this work. It has only a lower bound at 0, and lower values represent a higher performance. The following formulas describe the computation of this metric:

$$R_{i,j} = \frac{\frac{1}{n_i}\sum_{x \in C_i} d(x_i - \mu_i) + \frac{1}{n_j}\sum_{x \in C_j} d(x_j - \mu_j)}{d(\mu_i, \mu_j)} \quad where\ j\ is\ the\ most\ similar\ cluster\ to\ i \quad (6)$$

$$DB = \frac{1}{K}\sum_{\substack{i=1 \\ j=1 \\ i \neq j}}^{K} max\mathrm{R}_{i,j} \quad (7)$$

Here, R represents the similarity between clusters i and j, s_i is the mean of all distances between the points of cluster i and its centroid, $d_{i,j}$ is the distance between clusters i and j given by their centroids, and $max(R_{i,j})$ is the maximum similarity of clusters i and j.
DB is sensitive to the choice of distance metric and to overlapping clusters [18].

Calinski-Harabasz index (CH) [60], [61], or Variance Ratio Criterion, is computed as the ratio between the intra-cluster to inter-cluster dispersion. The dispersion is based on the sum of squared distances. For this metric, a higher value indicates a better result and it has no upper bound. The following formula describes the computation of this metric:

$$CH = \frac{tr(Bk)}{tr(Wk)} * \frac{n-k}{k-1} \quad (8)$$

Wk = SSE
Bk = SSB
the cluster's closeness or compactness is measured based on the distance between the cluster's centroid and the data points within the cluster while the cluster's separation from other clusters is measured using the distance from the cluster's centroid to the global centroid

where (Sw) is the intra-cluster scatter matrix, (SB) is the inter-cluster scatter matrix
Here, $tr(X)$ is the trace of the dispersion matrix (either between Bk or within Wk), n is the dataset size and k is the number of clusters.
CH is sensitive to irregular cluster shapes and sizes and to outliers [18].

Silhouette index (S) [60], [62] is computed as the ratio between the mean distance between a point and the rest of the points of that cluster and the mean distance between the point and all the points of the nearest cluster. SS has an interval of [-1, 1] where 1 represents well-separated dense clusters, 0 overlapping clusters, and -1 an incorrect clustering. Thus, SS evaluates as correct (and outputs higher scores for) the traditional structure of clusters. The following formula describes the computation of this metric:

$$a_i = \frac{1}{n_i} \sum_{\substack{x_i, x_j \in C_i \\ i \neq j}} d(x_i, x_j)$$

$$b_i = \min_{i!=j} \left(\frac{1}{n_j} \sum_{\substack{x_i \in C_i \\ x_j \in C_j}} d(x_i, x_j) \right)$$

$$S = \frac{1}{N} \sum_{i=1}^{N} \frac{b_i - a_i}{\max(b_i, a_i)}$$

Here, b is the mean of all distances between a point in cluster i and all points of the closest cluster j, and a is the mean of all distances between a point in cluster i and all other points in the same cluster.
S index can be sensitive to noise and outliers [18].

CVI library [67]

https://cluster-validity-indices.readthedocs.io/en/latest/_autosummary/cvi.modules.html

Xie-Beni index (XB) [66] evaluates the compactness and separation of a clustering solution by comparing total within-cluster compactness to the closest inter-cluster separation. It was originally developed for fuzzy clustering but can be adapted for hard partitions by setting the memberships to 1 for a point's assigned cluster and 0 otherwise. The index favors clusterings that produce small within-cluster squared errors while keeping cluster centres well separated. Lower XB values indicate better clustering solutions (more compact clusters and larger separation). The hard-partition XB is computed as:

$$XB = \frac{\sum_{k=1}^{K} \sum_{x \in C_k} d(x_i, \mu_k)^2}{N * \min_{1 \leq k < l \leq K} d(\mu_k, \mu_l)^2}$$

$$XB = \frac{\sum_{k=1}^{K}\sum_{i=1}^{N} {u_{ki}}^{m} * d(x_i, \mu_k)^2}{N * \min\limits_{1 \le k < l \le K} d(\mu_k, \mu_l)^2}$$

The numerator: total fuzzy within-cluster squared error (weighted by membership degrees u_{ij}^{m}).

XB does not consider the shape or density of clusters, it is sensitive to the number of clusters and its reliance on the Euclidian distance makes it susceptible to the curse of dimensionality [18].

the cluster cohesion is measured using the global mean squared distance of objects from the centroid of their cluster while the inter-cluster separation is measured using the minimum squared distance between pairs of clusters

Centroid-based Silhouette index (cSIL) [62], [75] is the cluster-level analogue of SS that uses centroids and mean squared distances instead of pairwise sample distances. For each cluster it compares the cluster's average squared within-cluster dispersion to the average squared distance from that cluster's points to the nearest other centroid, producing one silhouette score per cluster which are then averaged. cSIL ranges in [−1,1]: values near 1 indicate well-separated, compact clusters (under the centroid/squared-distance model), values near 0 indicate overlapping or ambiguous clusters, and values near −1 indicate poor clustering (clusters that are closer to other cluster centroids than to their own centroid)

$$a_k = \frac{1}{n_k} \sum_{x_k \in C_k} d(x_k, \mu_k)^2$$

$$b_k = \min_{k \neq l}((\frac{1}{n_l} \sum_{x_l \in C_l} d(x_l, \mu_l)^2) - d(\mu_k, \mu_l)^2)$$

$$cSIL = \frac{1}{K}\sum_{k=1}^{K} \frac{b_k - a_k}{\max\,(b_k, a_k)}$$

Generalized Dunn's index 43 (GD43) [69], [76] measures cluster quality as the ratio of the closest centre-to-centre separation to the worst (largest) cluster dispersion, where dispersion uses the half-power intra-cluster distances. GD43 is computed as the ratio between the minimum inter-cluster separation (here measured by centroid distance) and the maximum intra-cluster dispersion (a D3-style spread). GD43 is non-negative (no fixed upper bound) where larger values indicate better separation and compactness (values near 0 mean poor separation or a very dispersed cluster). Thus, GD43 rewards partitions whose closest cluster pair is still far relative to the worst cluster spread.

$$GD43 = \frac{\min\limits_{1 \le i < j \le K} d(\mu_i, \mu_j)}{\max\limits_{1 \le k \le K} \frac{2 * \sum_{x \in C_k} d(x, \mu_k)^{\frac{1}{2}}}{n_k}}$$

where *N* is the number of data points, *d(i,j)* is the Euclidean distance between data points x_i and x_j, and *max{d(i,k), k!=j}* is the maximum distance between data point *i* and any other data point in a different cluster.

Generalized Dunn's index 53 (GD53) [69], [76] is similar in spirit but replaces the centre-to-centre separation with a size-weighted average half-power distance between two clusters. As with GD43, higher GD53 values indicate better clustering. GD53 can be more robust than GD43 for non-spherical or differently sized clusters because it uses per-cluster half-power averages in the numerator rather than raw centre distances:

$$GD53 = \frac{\min\limits_{1 \le i < j \le K} \frac{\sum_{x \in C_i} d(x - \mu_i)^{\frac{1}{2}} + \sum_{x \in C_j} d(x - \mu_j)^{\frac{1}{2}}}{n_i + n_j}}{\max\limits_{1 \le k \le K} \frac{2 * \sum_{x \in C_k} d(x - \mu_k)^{\frac{1}{2}}}{n_k}}$$

Partition Separation (PS) [77] comprises a measure of separation between prototypes / centroids. Larger values of PS indicate better clustering solutions (maximization). PS is computed from prototype (centroid) separations adjusted by cluster size/balance (a prototype-distance based score, with fuzzy and hard variants). PS has no fixed interval (scale depends on data) and higher PS means prototypes are farther apart and cluster sizes more balanced. Thus, PS evaluates prototype separation (not internal compactness) and outputs higher scores for well-spaced, balanced prototype partitions.

$$PS = \sum_{k=1}^{K} \left(\frac{n_k}{\max\limits_{k \in K}(n_k)} - \exp\left[\frac{\min\limits_{1 \le i < j \le K} d(\mu_i, \mu_j)^2}{\frac{1}{K} \sum_{k=1}^{K} d(\mu_k, \mu_{data})^2} \right] \right)$$

Renyi's representative Cross Information Potential (rCIP) [78] represents each cluster C_k by a Gaussian centered at the cluster centroid c_k with a (regularized) sample covariance S_k, then measures the pairwise *overlap* of those Gaussians. rCIP aggregates the overlaps between all

distinct cluster pairs: small overlap means well-separated clusters. rCIP is non-negative (0 = perfect separation) and lower values indicate better clustering. The following formulas describe its computation:

$$S_k = \frac{1}{n_k - 1}\left(\sum_{x \epsilon C_k} xx^T - n_k \mu_k \mu_k^T\right) + \varepsilon$$

$$rCIP = \sum_{1 \le i < j \le K} (2\pi)^{-\frac{d}{2}} * \left|S_i + S_j\right|^{-\frac{1}{2}} * e^{-\frac{1}{2}(\mu_i - \mu j)^T (S_i + S_j)^{-1}(\mu_i - \mu_j)}$$

WB-index (WB) [79] measures how good a clustering result is by balancing within-cluster compactness and between-cluster separation. Lower values generally indicate better clustering quality. WB is computed as a scaled ratio of total within-cluster sum-of-squares to between-cluster dispersion (within / between SS combination, often multiplied or divided by K). WB is positive and smaller values indicate better partitions (it typically exhibits a minimum at the appropriate number of clusters). Thus, WB rewards compact, well-separated clusters (and is commonly used to select K by finding the minimum index).

WB considers average distances and not the shape or distribution of clusters, yet WB found as one of the most robust CVIs in a recent study [18].

$$WB = K * \frac{\sum_{k=1}^{K} \sum_{x \epsilon C_k} d(x, \mu_k)^2}{\sum_{k=1}^{K} n_k * d(\mu_k, \mu_{data})^2}$$

Permetrics library [65]

DBCV [25] is particularly designed for evaluating density-based clustering algorithms such as DBSCAN [123] The DBCV metric ranges from -1 to 1, with values close to 1 indicating a well-separated cohesive clusters (in terms of density), a value of 0 indicates an ambiguous structure, while near -1 values indicate poor cluster structures. The DBCV metric is calculated using the following formula:

All-points-core-distance (APCD) of an object $x \in C_k$ where KNN(x, i) is the distance between x and its i-th nearest neighbor in cluster k.

$$APCD(x) = \left(\frac{1}{n_k - 1} * \sum_{\substack{i=2 \\ x \in C_k}}^{n_k} \left(\frac{1}{KNN(x,i)} \right)^D \right)^{-\frac{1}{D}}$$

The mutual reacheability distance (MRD) between two object xi and xj

$$MRD(x_i, x_j) = \max \{APCD(x_i), APCD(x_j), d(x_i, x_j)\}$$

Minimum spanning trees (MST) are built for each cluster using MRDs. The density sparseness of a cluster (DSC), while density sparseness between clusters (DSBC) Ci and Cj can be defined as:

$$DSC(C_i) = \max \{weight\ of\ internal\ edges\ in\ MST\ of\ C_i\}$$

$$DSBC(C_i, C_j) = \min_{i \neq j} \{MRD(x_i, x_j) | x_i \in C_i, x_j \in C_j\}$$

$$DBCV(C_i) = \frac{\min_{i \neq j}(DSBC(C_i, C_j) - DSC(C_i))}{\max\left(\min_{i \neq j}(DSBC(C_i, C_j) - DSC(C_i))\right)}$$

$$DBCV = \sum_{k=1}^{K} \frac{n_k}{N} * DBCV(C_k)$$

DBCV also aims at density based clustering validation. It evaluates the clustering results based on the lowest density region inside a cluster and the highest density region between pairs of clusters. The DBCV index is formulated based on the all-points-core-distance (aptscoredist) which is the inverse of the density of each data point with respect to all other points inside its cluster. [27]

Although this index takes into account density and shape properties, it fails to handle datasets with not well separated clusters (datasets containing bridge between clusters) [124]. Although they show that DBCV outperforms all other metrics they analyzed (besides proposed) [124].

Dunn index (D) [69], [70] aims to quantify the compactness and separation between clusters in a clustering solution by considering both the distance between points within the same cluster (intra-cluster distance) and the distance between points in different clusters (inter-cluster distance). Specifically. D is the ratio between the minimum distance between any two clusters and the maximum distance found within any cluster. A higher D value indicates better clustering quality,

indicating that the clusters are well separated from each other while being compact internally; conversely, a lower D value may indicate that the clusters are too spread out or not well separated. The D index can be sensitive to the data scale and to outliers [18]. The following formulas allow for the computation of this index:

$$D(C_k) = \max_{x_i, x_j \in C_k,\ i \neq j} d(x_i, x_j)$$

$$D(C_k, C_l) = \min_{x_i \in C_k, x_j \in C_l} d(x_i, x_j)$$

$$D = \frac{\min_{1 \leq k < l \leq K} D(C_k, C_l)}{\max_{k \in [1,K]} D(C_k)}$$

Where the dataset is divided into Kclusters $C_1, C_2, \ldots, C_K$, and $d(x_i, x_j)$ denotes the Euclidean distance between points $x_i$and x_j.

Duda Hart index (DH) [72] is defined as the ratio between the average pairwise distance within clusters and the average pairwise distance between clusters. A lower value of the DHI indicates better clustering, indicating that the clusters are more compact and well-separated. The DHI assumes Euclidian distances, but it can be implemented to use other suitable distance metrics based on the specific problem and data characteristics. The following formulas allow for the computation of DHI:

$$D_{intra}(C_k) = \frac{2}{n_k(n_k - 1)} \sum_{x_i, x_j \in C_k} d(x_i, x_j)\ , where\ i < j$$

$$D_{inter}(C_k, C_l) = \frac{2}{n_k n_l} \sum_{x_k \in C_k} \sum_{x_l \in C_l} d(x_k, x_l)\ , where\ k \neq l$$

$$DH = \frac{\frac{1}{K} * \sum_{k=1}^{K} D_{intra}(C_k)}{\frac{2}{K(K-1)} * \sum_{k=1}^{K} \sum_{l=k+1}^{K} D_{inter}(C_k, C_l)}$$

Where the dataset is divided into Kclusters $C_1, C_2, \ldots, C_K$, each cluster C_k has n_k points, and $d(x_i, x_j)$ denotes the Euclidean distance between points $x_i$and x_j.

Sum of Squared Error index (SSE) [71] measures the sum of squared distances between each data point and its corresponding centroid or cluster center quantifying the compactness of the clusters. SSEI assigns each data point to its nearest centroid and calculates the squared Euclidean distance between the data point and its assigned centroid, the final value is obtained as the sum of the

squared distances for all data points. Higher SSE values indicate higher dispersion or greater variance within the clusters, while lower SSE values indicate more compact and well-separated clusters. The following formulas allow for the computation of SSE:

R-Squared index (RS) is based on the idea of comparing the variance of the data before and after clustering. The RS measures the proportion of the total variance in the data that is explained by the clustering solution. The R-Squared index ranges from -inf to 1, with higher values indicating better clustering solutions. A negative value indicates that the clustering solution is worse than random, while a value of 0 indicates that the clustering solution explains no variance beyond chance. The following formulas allow for the computation of RS:

$$RS = 1 - \frac{SSE}{SSE + BSS}$$

Where the dataset is divided into Kclusters $C_1, C_2, \dots, C_K$, each cluster C_k has n_k points, and $d(x_i, x_j)$ denotes the Euclidean distance between points $x_i$and x_j.

RS assumes a normal distribution [18].

Beale index (B) [63], [73], also known as the "variance ratio criterion" or the "F-ratio", measures the quality of a clustering solution by computing the ratio of the within-cluster sum of squares to the between-cluster sum of squares. The within-cluster sum of squares measures the variability of within each cluster, while the between-cluster sum of squares measures the variability between the clusters. A good clustering solution should have low within-cluster variation and high between-cluster variation, which results in a high B value. The B ranges from 0 to infinity, with higher values indicating better clustering solutions. However, B has a tendency to favor solutions with more clusters. The B can be calculated using the following formula:

$$B = \frac{\frac{BSS}{K-1}}{\frac{SSE}{N-K}}$$

Where the dataset containing *N* data points is divided into Kclusters $C_1, C_2, \dots, C_K$, each cluster C_k has n_k points, and $d(x_i, x_j)$ denotes the Euclidean distance between points $x_i$and x_j.

Ball Hall index (BH) [63], [64] computes the average distance between each data point and its cluster centroid and then averages this across all clusters. It measures the compactness and separation of clusters in a clustering result. A lower BH value indicates better clustering, as it signifies that the data points are closer to their own cluster centroid than to the centroids of other clusters, indicating a clear separation between clusters. The following formula allows for the computation of BH:

$$BH = \frac{1}{N} * \sum_{k=1}^{K} \frac{1}{2 * n_k} * \sum_{x \in C_k} d(x, \mu_k)$$

Where the dataset containing *N* data points is divided into Kclusters $C_1, C_2, \dots, C_K$, each cluster C_k has n_k points, and $d(x_i, x_j)$ denotes the Euclidean distance between points $x_i$and x_j.

Hartigan index (H) [74] also known as the Hartigan's criterion, is a measure used for evaluating the quality of clustering solutions. It is specifically designed for assessing the goodness of fit of a clustering algorithm, particularly the k-means algorithm. Lower values of the Hartigan index indicate better clustering solutions with lower within-cluster variance and higher separation between clusters. The Hartigan index can be calculated using the following formula:

$$H = \frac{\sum_{k=1}^{K} \sum_{x \epsilon C_k} d(x, \mu_k)^2}{\sum_{k=1}^{K} \sum_{x \epsilon C_k} d(x, \mu_l)^2}$$

Where the dataset containing *N* data points is divided into Kclusters $C_1, C_2, \dots, C_K$, each cluster C_k has n_k points and the a cluster center / centroid μ_k, and $d(x_i, x_j)$ denotes the Euclidean distance between points $x_i$and x_j, while in this case μ_l represents the centroid of the closest distinct cluster.

PyCVI library [68]

Score Function (SF) [80] is computed from two aggregated distances: a between-cluster term that measures how far cluster centroids lie from the global centroid, and a within-cluster term that measures average point-to-centroid scatter inside clusters. SF maps their difference through a double-exponential squashing function so that larger between-class separation and smaller within-class spread produce higher scores. SF has an interval of $(0,1)$, where values close to 1 indicate well-separated, compact clusters, values near 0 indicate poor or overlapping clusters. The following formula describes the computation of this metric:

$$BCD = \frac{1}{N * K}\sum_{k=1}^{K} n_k * d(\mu_k, \mu_{data})$$

$$WCD = \sum_{k=1}^{K}(\frac{1}{n_k} * \sum_{x_k \in C_k} d(x_k, \mu_k))$$

$$SF = 1 - \frac{1}{e^{e^{(BCD-WCD)}}}$$

estimates cluster centroids 'distances from the global centroids to evaluate the dispersion of clusters from each other. It also evaluates the clusters' degree of closeness by measuring the distance between the data objects and their respective cluster centroids.

SD [81] is computed as a combination of two complementary terms: the average scattering of clusters, which measures intra-cluster compactness, and the total separation, which quantifies the distances between cluster centroids. Specifically, SD evaluates the trade-off between how compact the clusters are and how well separated they are from each other. Lower SD values indicate compact and well-separated clusters, while higher values denote overlapping or dispersed structures. Thus, SD assesses clustering quality by considering high separation and low internal variance as a better clustering. SD is unable to correctly evaluate clusters with arbitrary shapes [25]. The following formulas describe the computation of this metric:

$$S = \frac{1}{K}\sum_{k=1}^{K} \frac{\sqrt{\sum_{d=1}^{D}(\sigma_d^k)^2}}{\sqrt{\sum_{d=1}^{D}(\sigma_d)^2}}$$

$$where\ \sigma_d\ is\ the\ variance\ on\ dimension\ d\ and\ \sigma_d^k\ is\ the\ variance\ of\ cluster\ k$$

$$\sigma_{c_k}^d = \sum_{i=1}^{n_k} \frac{(x_i^d - \mu_k^d)^2}{n_k}$$

$$D = \frac{\max\limits_{k \neq l} d(\mu_k, \mu_l)}{\min\limits_{k \neq l} d(\mu_k, \mu_l)} * \sum_{k=1}^{K} \frac{1}{\sum_{\substack{l=1 \\ k \neq l}}^{K} d(\mu_k, \mu_l)}$$

$$SD = \alpha * S + D$$

$$where\ \alpha = D(Kmax), but\ Kmax\ only\ for\ multiple\ analysis, so\ I\ guess\ K\ which\ is\ D$$

SDbw [82] evaluates clustering quality based on both cluster compactness and inter-cluster density separation. It combines two complementary components: *scattering*, which measures how compact clusters are relative to the overall dataset, and *density between-within*, which measures how much density exists between clusters compared to within them. Lower SDbw values indicate compact and well-separated clusters, while higher values correspond to overlapping or poorly separated structures. The following formulas describe the computation of this metric

$$stddev_k = \sqrt{\frac{1}{n_k} \sum_{x_k \in C_k} d(\mu_k, x_k)^2}$$

$$f(d) = 1 \; if \; d \leq stddev_k \; else \; 0$$

$$density(x, k) = \frac{1}{n_k} \sum_{x_k \epsilon C_k} f(d(\mu_k, x_k))$$

$$M_{kl} = \frac{\mu_k + \mu_l}{2}$$

the middle point of the line segment defined by the clusters' centers

$$Dbw = \frac{2}{K(K-1)} \sum_{k=1}^{K-1} \sum_{l=k+1}^{K} \frac{density(M_{kl}, k) + density(M_{kl}, l)}{\max\left(density(\mu_k, k), density(\mu_l, l)\right)}$$

$$SDbw = \alpha * S + Dbw$$

SDbw is sensitive to overlapping clusters and outliers [18] and is unable to correctly evaluate clusters with arbitrary shapes [25].

Xie-Beni* index (XB*) [52] is a normalized variant of the original Xie-Beni index designed to reduce sensitivity to data scale and the number of clusters. Like XB, it assesses clustering quality by balancing cluster compactness against inter-cluster separation, but XB* utilizes the maximum per-cluster average dispersion (it penalizes solutions that have one bad (loose/noisy) cluster even if others are tight - "bottleneck" style measure: the clustering is only as good as its worst cluster). It can be applied to both fuzzy and hard clusterings by defining membership degrees appropriately (1 for the assigned cluster, 0 otherwise in the hard case). The index rewards solutions with tightly grouped data points and well-separated cluster centers. Lower values correspond to higher quality clusterings. The hard-partition XB* is computed as

$$XB^* = \frac{\max\limits_{k \in K} \frac{\sum_{x \in C_k} d(x_i, \mu_k)^2}{n_k}}{\min\limits_{1 \leq k < l \leq K} d(\mu_k, \mu_l)^2}$$

$$XB^* = \frac{\max_{k \in K} \frac{\sum_{i=1}^{N} {u_{ki}}^m * d(x_i, \mu_k)^2}{n_k}}{\min_{1 \le k < l \le K} d(\mu_k, \mu_l)^2}$$

COP [84] measures the ratio between intra-cluster compactness (average distance from points to their cluster centroid) and a "farthest-neighbour" inter-cluster distance (for each cluster, the minimum over outside points of the maximum distance from that outside point to members of the cluster). COP takes values in $[0, \infty)$; values close to 0 indicate tight, well-separated clusters while large values indicate loose or poorly separated clusters. Thus, COP evaluates partitions that have small within-cluster scatter and relatively large farthest-neighbour separation.

$$COP = \frac{1}{K} \sum_{k=1}^{K} \frac{\frac{1}{n_k} \sum_{x_k \in C_k} d(x_k, \mu_k)}{\min_{x \notin C_k} \max_{y \in C_k} d(x, y)}$$

C index [83]

can only be used for evaluating the validity of clustering results obtained using the k-means algorithm due to its assumption that the clusters are spherical, equally sized, and have the same density. It is sensitive to the scale of the data, presence of noise and overlapping clusters.

$$S = \sum_{k=1}^{K} \sum_{\substack{i<j \\ x_i, x_j \in C_k}} d(x_i, x_j)$$

$$S_{min} = \sum_{r=1}^{m} d_{(r)}, m = \sum_{k=1}^{K} \binom{n_k}{2}, d_{(x)} \; sorted \; pairwise \; distances$$

$$S_{max} = \sum_{r=T-m+1}^{T} d_{(r)}, m = \sum_{k=1}^{K} \binom{n_k}{2}, T = \binom{N}{2}, d_{(x)} \; sorted \; pairwise \; distances$$

$$C = \frac{S - S_{min}}{S_{max} - S_{min}}$$

I index [55]

assumes that the clusters are convex and isotropic, which may not always be true in real-world datasets. It requires a large number of trials to find the optimal set of clustering parameters, which can be timeconsuming. It is not effective for datasets with skewed or imbalanced distributions, as it tends to produce similar scores for both wellclustered and poorlyclustered datasets.

$$I = \frac{\sum_{i=1}^{N} d(x_i, \mu_{data})}{SSE} * \max_{i \neq j} d(\mu_i, \mu_j)$$

CDbw [42] combines an internal-compactness measure with a separation measure that is penalized by density between clusters. Compactness: fraction of each cluster's points lying within a neighborhood radius $\sigma$of its centroid (averaged). Separation: typical nearest-centroid distance, reduced when there is measurable data density in the midpoint region between centroids. The final score multiplies separation by compactness (squared in your centroid simplification), so high CDbw favors clusterings that are both tight internally and well separated with low inter-cluster density. Higher values indicate better cluster structure.

It is hard for (CDbw) to find the representatives for each cluster, which makes the result of (CDbw) unstable [2], [19]

CDbw might be the most employed index for density based validation. It handles arbitrarily shaped clusters by representing each cluster with a number of points rather than one single representative point. The CDbw index evaluates the clustering results based on the inner-cluster cohesion and the inter-cluster separation [27]

Although this index has better performance in efficiently handling arbitrarily shaped clusters (by evaluating the density distribution within and between clusters) compared to other indices, it has some weakness and disadvantages related to the multiple representative points, including: – The use of the same number of representative points for all clusters is unfavorable, for a dataset with clusters of different shapes, densities and sizes such as the compound dataset of Fig. 1. – The pre-determination of the number of representative points, a parameter that significantly influences the performance of this index. – Even if an appropriate number of representative points can be defined, the representative points themselves can be determined by different approaches and thus the CDbw index can produce different evaluations. – CDbw index can be used for evaluation of datasets with well-separated clusters but may fail in the case of clusters connected with bridge or clusters with high-density area between them (high degree of overlap between clusters) such as the compound and path-based datasets of Fig. 1. [124]

$$RCR_{kl} = \{(\mu_k, \mu_l)\}$$

$$\mu_{kl} = \frac{1}{2}(\mu_k + \mu_l)$$

$$f(x,u) = 1 \; if \; d(x,u) < \; \sigma \; else \; 0$$

$$card(\mu_k) = \frac{1}{n_k} \sum_{x \in C_k} f(x, \mu_k)$$

$$card(\mu_{kl}) = \frac{1}{n_k + n_l} \sum_{x \in C_k \cup C_l} f(x, \mu_{kl})$$

$$Dens(C_k, C_l) = \frac{d(\mu_k, \mu_l)}{2\sigma} * card(\mu_{kl})$$

$$InterDens = \frac{1}{K} \sum_{k=1}^{K} \max_{l \neq k} Dens(C_k, C_l)$$

$$Separation = \frac{1}{K} \sum_{k=1}^{K} \frac{\min_{l \neq k} d(\mu_k, \mu_l)}{1 + InterDens}$$

$$Compactness = \frac{1}{K} \sum_{k=1}^{K} card(\mu_k)$$

$$Cohesion = \frac{Compactness}{1 + IntraChange}$$

$$CDbw = Cohesion * Separation * Compactness$$

CDbw is, as far as we know, the most employed relative measure for densitybased validation. The approach adopted by CDbw is to consider multiple representative points per cluster, instead of one, and thereby to capture the arbitrary shape of a cluster based on the spatial distribution of such points. CDbw has, however, several major drawbacks related to the multiple representatives it employs. The first of these drawbacks is how to determine the number of representative points for each cluster. Given that clusters of different sizes, densities and shapes are under evaluation, employing a fixed number of representatives for all clusters does not seem the best approach. Even if a single number of representative points is employed for all clusters (as the authors suggest), this number can still be critical to the performance of the measure and is a parameter, which is, at the very least, undesirable. Assuming that a reasonable number of representative points can be defined, the representative points themselves have to be determined. Different approaches can be employed in such a task for CDbw, as suggested by the authors. The adoption of different approaches to find representative points can not only be seen as another parameter, but as a significant source of instability, given that two different sets of representatives generated by different approaches, can lead to different evaluations. [25]

VIASCKDE index [46]

is an index that is not affected by the cluster shape, and thus, it can make a realistic evaluation of clustering performance regardless of the clusters' shape. Unlike the existing cluster validation indices, our index calculates the compactness and separation values of the cluster based on calculating the compactness and separation values for each data separately. In other words, it calculates the compactness and separation values of the cluster over the distance of data, independent of parameters such as the cluster center because, in non-spherical clusters, the distance of the data to the closest data is more important than its distance to the cluster center. As can be seen in the example given in figure below, the closest data in the cluster that "it belongs to" is used when calculating the compactness value for the data x. Similarly, the separation value of x is calculated by the distance to the closest data of the cluster that "it does not belong".

$$a(x) = \min_{i!=j}(\sum_{\substack{x\in C_i \\ y\in C_i}} d(x,y))$$

$$b(x) = \min_{i!=j}(\sum_{\substack{x\in C_i \\ y\in C_j}} d(x,y))$$

CoSeD (Compactness and Separation Value of a Data): the CoSeD can be described as the compactness and separation value of any data, we calculated the weight of each data that is W_{KDE} according to obtained KDE value

$$CoSeD(x) = W_{KDE} * \frac{b(x) - a(x)}{\max(b(x), a(x))}$$

$$CoSeD(C_k) = \frac{1}{n_k}\sum_{i=1}^{n_k} CoSeD(x_i)$$

$$VIASCKDE = \frac{\sum_{k=1}^{K} CoSeD(C_k)}{\sum_{k=1}^{K} n_k}$$

CS index [41] is similar in concept to the DB and D indices.

$$CS = \frac{\sum_{k=1}^{K}(\frac{1}{n_k} * \sum_{x_i\in C_k} \max_{x_j\in C_k} d(x_i, x_j))}{\sum_{k=1}^{K} \min_{\substack{l\in K \\ k\neq l}} d(\mu_k, \mu_l)}$$