Expert Systems With Applications 223 (2023) 119784

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A novel cluster validity index based on augmented non-shared nearest neighbors

Xinjie Duan, Yan Ma *, Yuqing Zhou, Hui Huang, Bin Wang

College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, PR China

ARTICLE INFO

Keywords:
Validity index
Within-cluster compactness
Between-cluster separation
Shared nearest neighbors

ABSTRACT

The true cluster number of the dataset in practical applications is rarely known in advance. Therefore, it is necessary to use a cluster validity index to evaluate the clustering results and determine the optimal cluster number. However, the performance of existing cluster validity indices is vulnerable to various factors such as cluster shape and density. To solve the above issues, this paper proposes a new cluster validity index based on augmented non-shared nearest neighbors (ANCV). The ANCV index is based on the following principles: (1) Within-cluster compactness can be measured by the distance between the pairs of data points with fewer shared nearest neighbors. (2) The distances between the pairs of data points at the intersection of clusters can be used to estimate the between-cluster separation. On this basis, the above point pairs are further extended to their augmented non-shared nearest neighbors, thereby forming small clusters. Then, the average distance within and between these clusters is calculated respectively to estimate the within-cluster compactness and between-cluster separation. Finally, the optimal number of clusters is determined by the difference between the between-cluster separation and the within-cluster compactness. Experimental results on both 12 two-dimensional synthetic datasets and 10 real datasets from UCI have shown that the ANCV index performs the best among all compared indices.

1. Introduction

In cluster analysis, objects are grouped into clusters so that those in the same cluster are more similar and those in different clusters are less similar. Cluster analysis has been widely applied in recent years to fields such as artificial intelligence, biomedicine, machine learning, and genetics. Many algorithms for clustering are based on finding the cluster centers, such as the K-means algorithm (Yang, Ma, Zhang, Li, & Zhang, 2017) and the density peak clustering (DPC) algorithm (Rodriguez & Laio, 2014). One of the main disadvantages of the K-means algorithm is that it cannot identify non-spherical datasets. Kernel k-means (X. Liu, 2022; X. Liu, et al., 2019) is an extension of standard K-means clustering that identifies non-spherical clusters by expressing the distance in the form of a kernel function. Clustering algorithms such as hierarchical clustering divide and merge clusters based on between-cluster distance (Pfeifer & Schimek, 2021). Other clustering algorithms, such as DBSCAN (Hahsler, Piekenbrock, & Doran, 2019), utilize the distribution of densities within and between clusters.

In addition, some algorithms represent data points as minimum spanning tree or nearest neighbor graph. The multi-stage hierarchical clustering algorithm (CTCEHC) uses the centroid of the minimum spanning tree to determine the cluster center (Ma, Lin, Wang, Huang, & He, 2021). The neighborhood-based three-stage hierarchical clustering algorithm (NTHC) performs clustering by shared nearest neighbors and 1-nearest neighbor (Wang, Ma, & Huang, 2021). Furthermore, a split-merge clustering algorithm based on the k-nearest neighbor graph (SMKNN) is proposed, where the KNN graph guides the clustering process (Wang, Ma, Huang, Wang, & Acharjya, 2023). The DDC algorithm uses densities decreased chains to cluster data of any shape and density (Li & Cai, 2022). While these algorithms are effective for non-spherical clusters, they all require the input of a cluster number, since the true cluster number is frequently unknown at the time of clustering. Therefore, the cluster validity indices (CVIs) are used to evaluate the clustering results for different cluster numbers and determine the optimal cluster number.

There are two types of CVIs: internal index and external index. The external index compares the clustering results with the true labels. There are three main categories of external validity indices: pair-counting, set-

* Corresponding author.
E-mail addresses: ma-yan@shnu.edu.cn (Y. Ma), huanghui@shnu.edu.cn (H. Huang), binwang@shnu.edu.cn (B. Wang).

https://doi.org/10.1016/j.eswa.2023.119784
Received 27 August 2022; Received in revised form 3 February 2023; Accepted 1 March 2023
Available online 11 March 2023

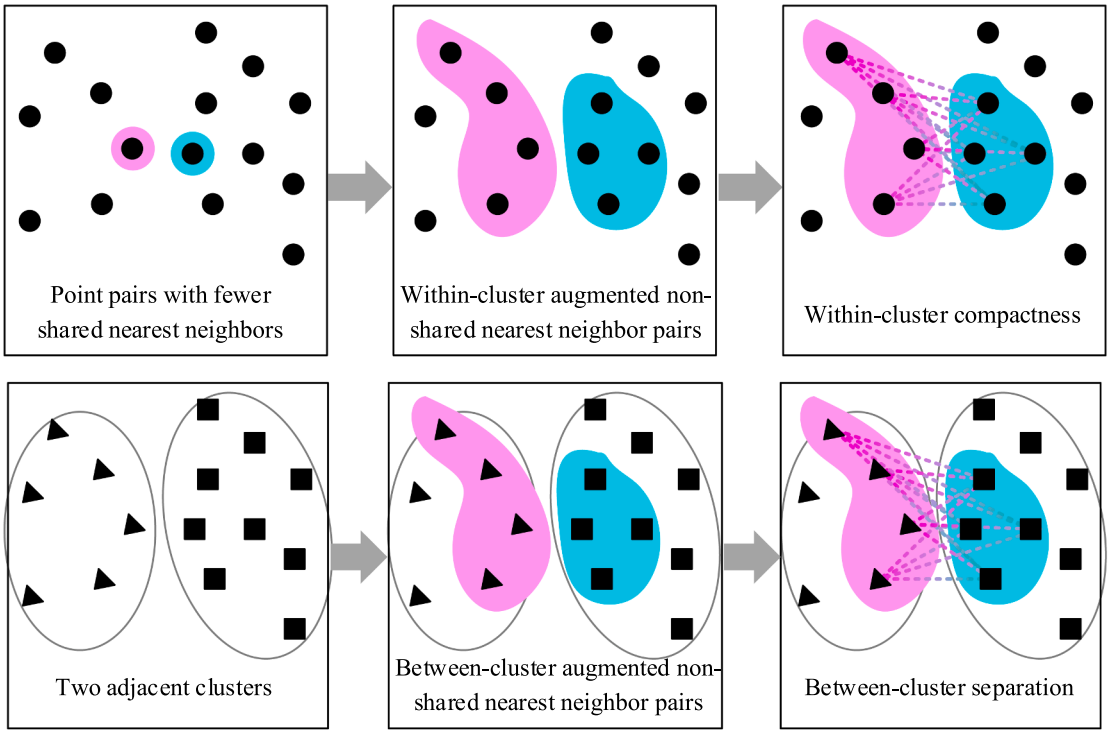
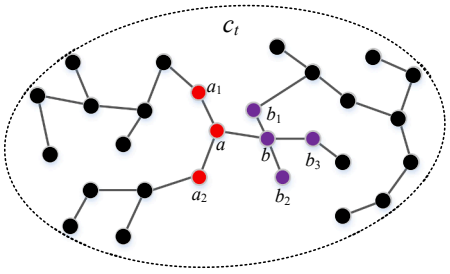
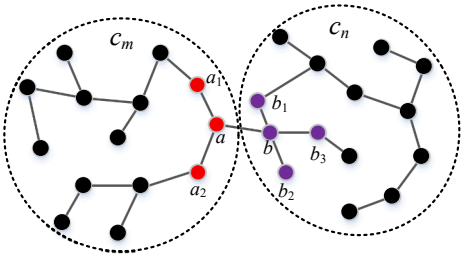


Fig. 1. A schematic diagram of the proposed algorithm.



(a) Within-cluster augmented non-shared nearest neighbor pairs



(b) Between-cluster augmented non-shared nearest neighbor pairs

Fig. 2. Within-cluster and between-cluster augmented non-shared nearest neighbor pairs.

matching, and information theory (van der Hoef & Warrens, 2019). The pair-counting measures count the number of point pairs that differ or agree between two clustering results. The Rand index (Rand, 1971) and the adjusted Rand index (ARI) (Hubert & Arabie, 1985) are two commonly used pair-counting measures. As opposed to comparing pairs of points, set-matching measures compare pairs of clusters. Examples of set-matching measures are F1-measure (F1) (de Souto, et al., 2012), Purity (Rendón, Abundez, Arizmendi, & Quiroz, 2011), and Centroid Ratio (CR) (Zhao & Fränti, 2013). Information-theoretic measures are used to determine how much information is shared between two partitions. In recent years, information-theoretic indices have become increasingly popular since they are based on strong mathematical foundations (Lei, et al., 2016; Shannon, 1948). The Entropy index measures the purity of the cluster class labels (Rendón, et al., 2011). In addition, information-theoretic indices include variations in information and different normalizations of mutual information (MI) (Meilă, 2007; Pfitzner, Leibbrandt, & Powers, 2009).

Unlike the external index, the internal index evaluates clustering results directly. Since the true labels of data points are often difficult to obtain, internal indices are better suited for verifying the validity of the clustering results. Common internal indices are the Davies-Bouldin index (DB) (Singh, Mittal, Malhotra, & Srivastava, 2020), Silhouette index (SIL) (Rousseeuw, 1987), COP (Gurrutxaga, et al., 2010), Calinski-Harabasz (CH) (Cengizler & Kerem-Un, 2017), and Dunn-index (Dunn, 1974), etc. These indices, however, are only applicable to spherical clusters. For example, the DB index uses the average value of the data points within a cluster as the center of the cluster. If the cluster center is chosen incorrectly, the optimal cluster number can be incorrect for arbitrarily shaped clusters.

Furthermore, some internal indices involve membership in fuzzy c-means clustering algorithm, such as PCAES (Wu & Yang, 2005) and IMI (Yun Liu, Jiang, Hou, & Liu, 2021). Compared to other indices, the PCAES index is less affected by noise, but more affected by the initial cluster centers. For unbalanced datasets, the IMI index performs well. As these indices are dependent on cluster centers, they will produce inaccurate results if the cluster center is incorrect. The SV and OS indices (Žalik & Žalik, 2011) solve this problem by calculating compactness and overlap measures based on a few data points in the cluster. However, the OS index performs significantly worse when there is an overlap between clusters. Aside from Euclidean distances, there are also indices based on point symmetry distances, such as the Sym index (Bandyopadhyay & Saha, 2008). The Sym index, however, is only applicable to datasets that are internally symmetric.

Several internal indices for arbitrarily shaped clusters have been

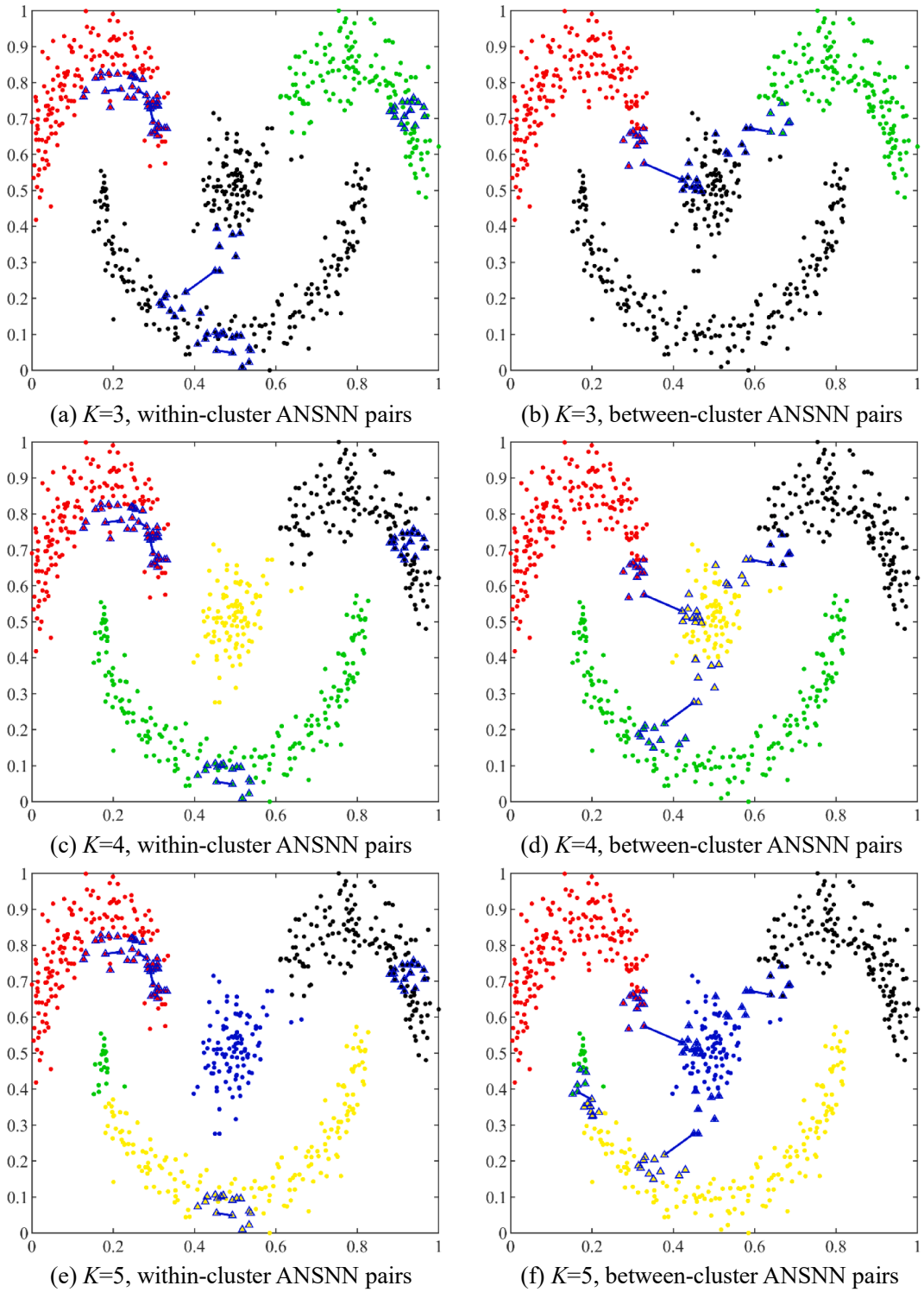


Fig. 3. Clustering results of the Smile Face dataset using the NTHC algorithm with blue triangles indicating the within-cluster and between-cluster augmented non-shared nearest neighbor pairs.

proposed in recent years. The STR index (Starczewski, 2017) uses knee point detection based on the DB index. The CVNN index (Yanchi Liu, et al., 2013) combines the idea of *k*-nearest neighbors to measure the between-cluster distance based on the nearest neighbor distribution of the data points. The BWC index (Zhou, Liu, & Song, 2021) and the LCCV index (Cheng, Zhu, Huang, Wu, & Yang, 2018) both suggest improvements to the SIL index. The former measures distances within and between clusters using the average distance between the center and the points within a cluster and the shortest distance between the centers of different clusters. The latter uses the concept of natural nearest neighbors to calculate the density kernels. And the between-cluster distance is determined by the geodesic distances between density kernels. The DCVI index (Xie, Xiong, Dai, Wang, & Zhang, 2020) constructs a minimum spanning tree for the density kernel and uses the minimum spanning tree to compute within-cluster and between-cluster distances. In addition, the SSDD index (Liang, Han, & Yang, 2020) evaluates the clustering

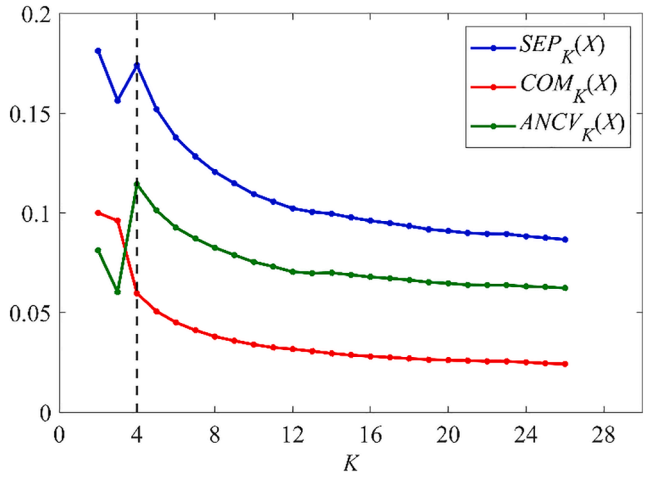


Fig. 4. The clustering evaluation results of the Smile Face dataset using the ANCV index, where K varies from 2 to 26 and the red, blue, and green lines represent the within-cluster compactness, between-cluster separation, and the ANCV index results, respectively.

results based on the variation in density between density kernels. These indices are effective for arbitrarily shaped clusters, however, they are less effective if the dataset contains clusters of varying densities.

In this paper, the proposed CVI is guided by the following principles: (1) The within-cluster compactness is related to the distance between the data point pairs with fewer shared nearest neighbors within a cluster. (2) The between-cluster separation can be measured by the distances between the non-shared nearest neighbor pairs located at the intersection of adjacent clusters. Accordingly, this paper proposes a new cluster validity index based on augmented non-shared nearest neighbors (ANCV). We first construct a minimum spanning tree according to the distance between points in the dataset. Secondly, we determine the within-cluster compactness and between-cluster separation by the distances between pairs of augmented non-shared nearest neighbors located within and between clusters, respectively. Lastly, the optimal number of clusters is determined by evaluating the difference between the between-cluster separation and the within-cluster compactness. Experimental results on both synthetic and real datasets have shown that the proposed index performs the best among all compared indices.

In this study, we aim to develop a cluster validity index whose performance is less affected by the density and shape of the clusters. Fig. 1 shows a schematic diagram of the proposed algorithm. We now briefly analyze the proposed algorithm in two aspects. (1) The within-cluster compactness is determined by the average distance between pairs of within-cluster augmented non-shared nearest neighbors, which are derived from point pairs with fewer shared nearest neighbors. In addition, point pairs with fewer shared nearest neighbors are determined by the distribution of local data points within a cluster rather than the shape and density of the entire cluster. (2) The between-cluster separation is determined by the average distance between pairs of between-cluster augmented non-shared nearest neighbors. These point pairs are derived based on the distribution of local data points between clusters regardless of the shape or density of the entire cluster. In conclusion, the proposed cluster validity index is effective for clusters of varying shapes and densities in the dataset.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of existing CVIs and their shortcomings. Section 3 describes the proposed index in this paper. Experimental results on synthetic and real datasets are given in Section 4. Section 5 discusses several factors that affect ANCV performance. In section 6, we present our conclusions and discuss future research directions.

2. Related work

In recent years, researchers have proposed a variety of different CVIs.

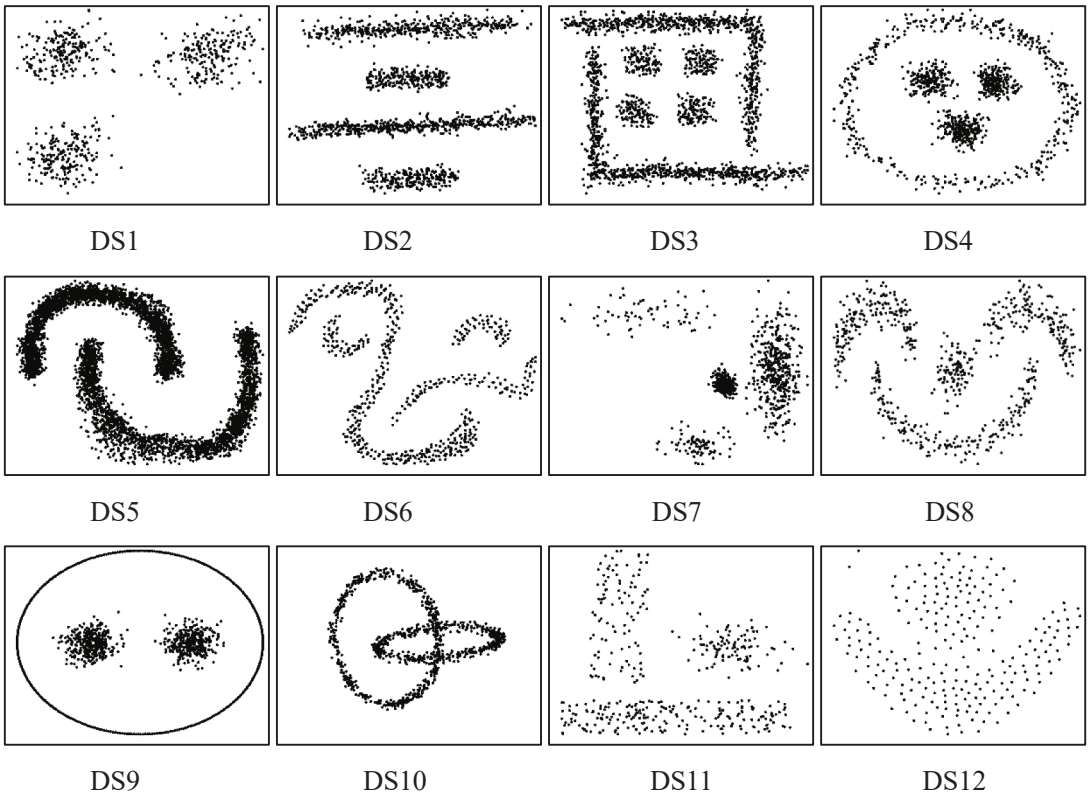


Fig. 5. 12 two-dimensional synthetic datasets.

Table 1
The description of 12 two-dimensional synthetic datasets.

Dataset	Data size	Cluster number	Spherical cluster	Ring-shaped cluster	Arc-shaped cluster	Linear cluster
DS1	600(200,200,200)	3	3			
DS2	1268(466,382,213,207)	4				4
DS3	1741(623,691,106, 103,113,105)	6	4			2
DS4	1015(407,187,224,197)	4	3	1		
DS5	4811(2520,2291)	2			2	
DS6	630(410,58,100,62)	4			4	
DS7	863(421,86,294,62)	4	4			
DS8	644(201,181,161,101)	4	1		3	
DS9	999(333,333,333)	3	2	1		
DS10	1000(500,500)	2		2		
DS11	400(200,100,100)	3	1			2
DS12	240(87,153)	2	1		1	

Table 2
The description of 10 real datasets.

Datasets	Abbreviations	Data size	Dimensionality	Number of clusters
Contraceptive	R1	1473	9	3
Connectionist	R2	208	61	2
German	R3	1000	24	2
Glass	R4	214	9	6
Leuk	R5	72	40	3
Pittsburg-bridges-REL-L	R6	103	7	3
Sonar	R7	208	60	2
Wifilocalization	R8	2000	7	4
Wilt	R9	4839	5	2
Zoo	R10	101	16	7

Traditionally, the clustering results are evaluated by using a single data point as a representative for the cluster. For example, the DB index uses the cluster center as a representative of the cluster. The minimum value of the DB index indicates the minimum within-cluster distance and the maximum between-cluster distance, which reflects the most reasonable partitioning of the dataset. The DB index is defined as follows:

$$DB(K) = \frac{1}{K}\sum_{i=1}^{K}\max_{i\neq j}\left[\frac{avg(C_i)+avg(C_j)}{d(v_i,v_j)}\right] \tag{1}$$

where K denotes the number of clusters, v_i and v_j are the average of all the data points in clusters C_i and C_j, respectively, and $avg(C_i)$ and $avg(C_j)$ denote the within-cluster distances of clusters C_i and C_j, respectively.

$$avg(C_i) = \frac{1}{|C_i|}\sum_{x_i\in C_i} d(x_i, v_i) \tag{2}$$

where $|C_i|$ denotes the number of data points in cluster C_i.

Similar to the DB index, the CH index takes the cluster centers and the center of the dataset as representative points, and measures the within-cluster and between-cluster distances by the squared sum of the distances from each point within a cluster to the cluster center and the squared sum of the distances between each cluster center and the center of the dataset, respectively. According to the Dunn index, the within-cluster and between-cluster distances are measured using the distance between the furthest and closest point pairs within a cluster and between clusters, respectively. In addition, the PBM index (Pakhira, Bandyopadhyay, & Maulik, 2004) introduces the membership degree based on the DB index for fuzzy clustering. The above method of describing clusters based on cluster centers is only appropriate for spherical clusters and is less effective for arbitrarily shaped clusters.

To adapt the indices to a variety of cluster shapes, the researchers selected multiple points from the cluster to serve as representative points. The data points that have at least one neighbor in the k-nearest neighbors belonging to different clusters are considered as representative points for the CVNN algorithm. And the size of the between-cluster distance depends on the weights of these representative points. The details are as follows:

$$Sep(NC,k) = \max_{i=1,2,\ldots,NC}\left(\frac{1}{n_i}\sum_{j=1}^{n_i}\frac{q_j}{k}\right) \tag{3}$$

where n_i is the number of data points in cluster C_i, k is the number of nearest neighbors, and q_j is the number of nearest neighbors in the k-nearest neighbors of the jth point in cluster C_i that are in different clusters.

The CVNN index defines within-cluster distance as the average distance between all point pairs within a cluster.

$$Com(NC) = \sum_{i=1}^{NC}\left[\frac{2}{n_i(n_i-1)}\sum_{x,y\in C_i} d(x,y)\right] \tag{4}$$

The CVNN index value is the sum of the normalized within-cluster and between-cluster distances:

$$CVNN(NC,k) = Sep_{NORM}(NC,k) + \mathrm{Com}_{\mathrm{NORM}}(\mathrm{NC}) \tag{5}$$

When there is no overlap between clusters, the number of representative points is small, so the value of *Sep* is smaller. In contrast, the value of *Sep* is higher when there is an overlap between clusters. When the CVNN index value reaches a minimum, it indicates that the dataset has been reasonably partitioned. Similar to CVNN, the SSDD index and RTI index (Rojas-Thomas, Santos, & Mora, 2017) also use multiple points in a cluster as representative points. In the SSDD index, the data points with greater densities are regarded as representative points. The minimum spanning tree containing representative points is used as the backbone of the cluster, and the clustering results are evaluated based on the variation of the densities of the data points on the backbone. According to the RTI index, each sub-cluster center is considered as a representative and the within-cluster and between-cluster distances are determined by the weight of the edges on the minimum spanning tree comprised of the representative points. Unlike the above indices, the LCCV and DCVI indices assign different size neighborhoods to each data point based on the idea of natural nearest neighbors and use denser points within the neighborhood as representative points. The LCCV index uses the geodesic distance between the representative points and evaluates the clustering results based on the SIL index. The DCVI index constructs a minimum spanning tree based on representative points and uses the ratio of the longest edge within a cluster and the shortest edge between clusters as the outcome. In general, the above indices can identify well-separated clusters of arbitrary shape, however, they do not take into account the boundary of the clusters and their performance is limited by the choice of representative points.

Table 3
The description of 13 indices.

Name	Optimal value	Definition	Reference
Dunn	Max	$\min_{1\leqslant i\leqslant K}\left\{\min_{1\leqslant j\leqslant K, i\neq j}\left(\min_{x\in C_i, y\in C_j} d(x,y)/\max_{1\leqslant k\leqslant K}\left\{\max_{a,b\in C_k} d(a,b)\right\}\right)\right\}$	Dunn (1974)
DB	Min	$\frac{1}{K}\sum_{i=1}^{K}\max_{1\leqslant j\leqslant K, i\neq j}\left\{\left[\frac{1}{n_i}\sum_{x\in C_i} d(x,v_i)+\frac{1}{n_j}\sum_{x\in C_j} d(x,v_j)\right]\Big/ d(v_i,v_j)\right\}$	Singh, et al. (2020)
CH	Max	$\frac{1}{K-1}\sum_{i=1}^{K} n_i d^2(v_i,v)/\frac{1}{N-K}\sum_{i=1}^{K}\sum_{x\in C_i} d^2(x,v_i)$	Cengizler and Kerem-Un (2017)
SIL	Max	$\frac{1}{K}\sum_{i=1}^{K}\left\{\frac{1}{n_i}\sum_{x\in C_i}[b(x)-a(x)]/\max[b(x),a(x)]\right\}$ $a(x)=\frac{1}{n_i-1}\sum_{y\in C_i, x\neq y} d(x,y), b(x)=\min_{1\leqslant j\leqslant K, i\neq j}\left[\frac{1}{n_j}\sum_{y\in C_j} d(x,y)\right]$	Rousseeuw (1987)
CVNN	Min	$\max_{1\leqslant i\leqslant K}\left(\frac{1}{n_i}\sum_{j=1}^{n_i}\frac{q_j}{k}\right)+\sum_{i=1}^{K}\left[\frac{2}{n_i(n_i-1)}\sum_{x,y\in C_i} d(x,y)\right]$	Yanchi Liu, et al. (2013)
DCVI	Min	$\sum_{i=1}^{K}\max_{e\in E_{T_i}}\{W(e)\}/\min_{1\leqslant j\leqslant K, i\neq j}\left\{\min_{x\in C_i, y\in C_j}(d(x,y))\right\}$	Xie, et al. (2020)
SSDD	Min	$\sum_{i=1}^{K}\left[\frac{\max(A_{C_i})-\min(A_{C_i})}{\max(A_{C_i})}+\frac{\mathrm{mean}(P_{C_i})}{\max\{\mathrm{mean}(P_{C_i}),\mathrm{mean}(A_{C_i})\}}\right]$	Liang, et al. (2020)
COP	Min	$\frac{1}{N}\sum_{C_i}\lvert C_i\rvert\frac{1/\lvert C_i\rvert\sum_{x_j\in C_i} d(x_j,v_i)}{\min_{x_j\notin C_i}\max_{x_k\in C_i} d(x_j,x_k)}$	Gurrutxaga, et al. (2010)
IMI	Min	$\frac{\sum_{i=1}^{K}\left(\frac{\sum_{j=1}^{N} u_{ij}^2\lVert x_j-v_i\rVert^2}{\sum_{j=1}^{N} u_{ij}}\right)}{\min_{i\neq k}\delta_{ik}\lVert v_i-v_k\rVert^2+\mathrm{median}_{i\neq k}\delta_{ik}\lVert v_i-v_k\rVert^2}, \delta_{ik}=\frac{\sum_{i=1}^{N} u_{ik}}{\sum_{i=1}^{N} u_{ij}}$	Yun Liu, et al. (2021)
OS	Min	$\frac{\sum_{C_i}\sum_{x_j\in C_i} ov(x_j,C_i)}{\sum_{C_i} 10/\lvert C_i\rvert\sum\max_{x_j\in C_i}(0.1\lvert C_i\rvert)\{d(x_j,v_i\}}$	Žalik and Žalik (2011)
PCAES	Max	$\sum_{i=1}^{K}\sum_{j=1}^{N} u_{ij}^2/u_M-\sum_{i=1}^{K}\exp\left(-\min_{k\neq i}\{\lVert x_i-x_k\rVert^2\}/\beta_T\right) u_M=\min_{1\leqslant i\leqslant K}\left\{\sum_{j=1}^{N} u_{ij}^2\right\},\ \beta_T=\frac{\sum_{l=1}^{K}\lVert v_l-\bar{v}\rVert^2}{K},\ \bar{v}=\sum_{i=1}^{K} v_i/K$	Wu and Yang (2005)
SV	Max	$\frac{\sum_{i=1}^{K}\min_{j\in[1,\ldots,K], i\neq j} d(v_i,v_j)}{\sum_{i=1}^{K}\max_{x_j\in C_i} d(x_j,v_i)}$	Žalik and Žalik (2011)
Sym	Max	$\max_{i,j=1}^{K}\lVert v_i-v_j\rVert/\left(K\sum_{i=1}^{K}\sum_{j=1}^{n_i} d_{ps}^{*}(\bar{x}_j^i,v_i)\right)$	Bandyopadhyay and Saha (2008)

K: number of clusters; N: number of the all data points in dataset; C_i: the ith cluster of dataset; n_i: number of data points in C_i; v_i: the average of all data points in cluster C_i; v: the average of all data points in dataset; $d(x,y)$: the Euclidean distance between data points x and y; A_{C_i}: the density of the region where two adjacent points on the backbone of cluster C_i are located; P_{C_i}: the density of region where the nearest data point pairs between clusters are located; E_{T_i}: the set of edges on the minimum spanning tree based on the representative points in cluster C_i; $W(e)$: the weight of edge e on the minimum spanning tree; u_{ik} is the membership value of x_i to v_k; $ov(\cdot)$ represents the overlap degree; d_{ps}^{*} measures the point symmetry between a data point and a cluster center.

3. Methodology

In this section, we describe and explain the definition, rationale, complexity, and parameters related to the ANCV index.

3.1. Definition

Given a dataset $X_{N\times D}=\{x_1,x_2,\cdots,x_N\}$ containing N data points of dimension D. Suppose that the clustering algorithm partitions the dataset X into K clusters, denoted as $C=\{c_1,c_2,\cdots,c_K\}$. And we further assume that $G_X=(V,E_X)$ is the complete graph of the dataset X,$T_X=(V,E_{T_X})$ is the minimum spanning tree constructed on G_X, where V is the set of vertices consisting of N data points, and E_{T_X} denotes the set of all edges on the minimum spanning tree. And the edges on T_X associated with data points x_i and x_j are weighted by $d(x_i,x_j)$ which is the Euclidean distance between data points x_i and x_j.

Definition 1. (***shared nearest neighbors***)

Let $N_k(i)$ and $N_k(j)$ be respectively the sets of k-nearest neighbors (kNN) of data points x_i and x_j in the dataset X. The set of shared nearest neighbors (SNN) of data points x_i and x_j is defined by

$$N_S(i,j)=N_k(i)\cap N_k(j) \tag{6}$$

Definition 2. (***non-shared nearest neighbors***)

Let $N_S(i,j)$ be the SNN set of data points x_i and x_j in the dataset X and $N_k(i)$ be the kNN set of data point x_i. The set of non-shared nearest neighbors (NSNN) of x_i with respect to x_j is defined by

$$N_{\bar{S}}(i)_j=N_K(i)-N_S(i,j) \tag{7}$$

According to Definition 2, the SNN of x_i and x_j are removed from the kNN set of data point x_i and the remaining points are called the non-shared nearest neighbors of x_i with respect to x_j. Similarly, the NSNN set of x_j with respect to x_i is defined by

$$N_{\bar{S}}(j)_i=N_K(j)-N_S(i,j) \tag{8}$$

Definition 3. (***augmented non-shared nearest neighbors***)

Let $N_{\bar{S}}(i)_j$ be the NSNN set of x_i with respect to x_j. The set of augmented non-shared nearest neighbors (ANSNN) of x_i with respect to x_j is defined by

$$\widetilde{N}_{\bar{S}}(i)_j=N_{\bar{S}}(i)_j\cup\{x_i\}-\{x_j\} \tag{9}$$

For the data points x_i and x_j in the dataset X, there must exist $x_i,x_j\notin N_S(i,j)$. Suppose $x_j\in N_k(i)$, then $x_j\in N_{\bar{S}}(i)_j$. According to Definition 3, x_j is removed from $N_{\bar{S}}(i)_j$ such that $x_j\notin\widetilde{N}_{\bar{S}}(i)_j$.

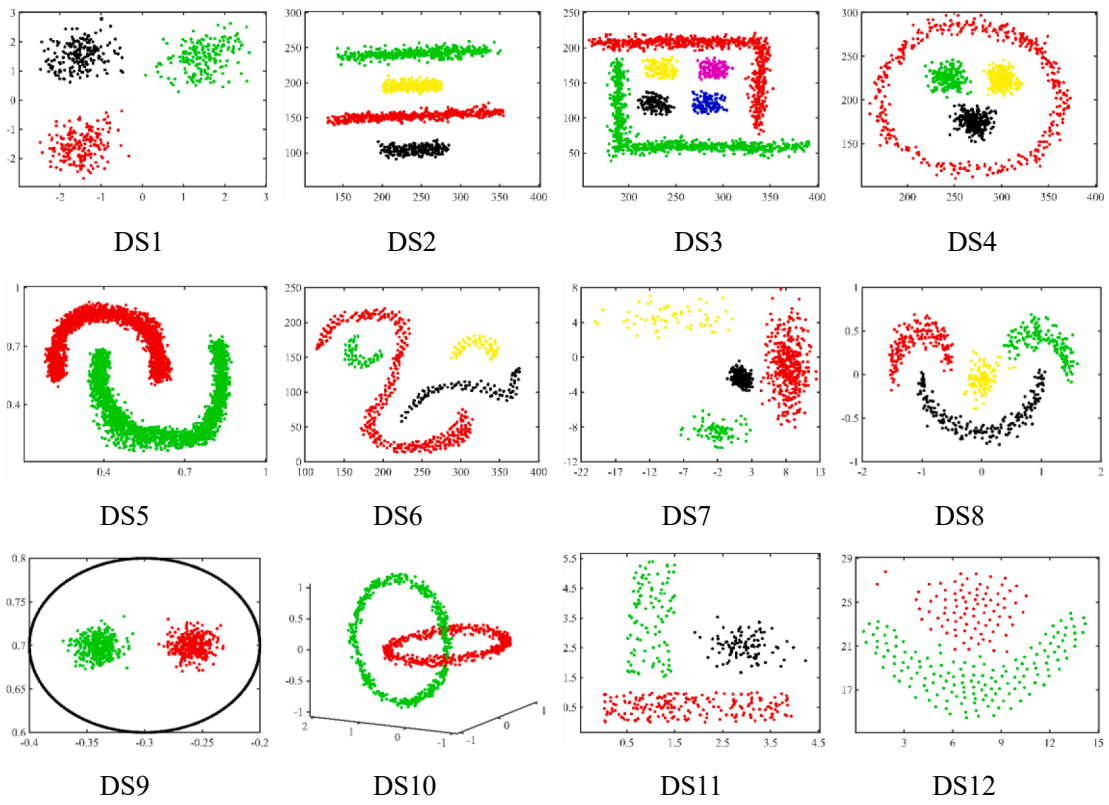


Fig. 6. Clustering results on 12 synthetic datasets using CTCEHC under true cluster number.

Table 4
The results on 12 two-dimensional synthetic datasets.

Datasets	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS9	DS10	DS11	DS12	N_{hit}
T^*	3	4	6	4	2	4	4	4	3	2	3	2	—
Dunn	2	2	5	2	**2**	3	5	3	2	**2**	**3**	12	3
DB	**3**	8	19	19	12	12	6	6	12	19	4	4	1
CH	**3**	26	23	15	25	26	7	20	12	32	6	4	1
SIL	**3**	7	22	10	10	9	6	6	12	19	6	4	1
CVNN	**3**	7	8	**4**	12	5	6	7	**3**	11	4	4	3
DCVI	**3**	**4**	**6**	**4**	**2**	**4**	21	3	**3**	**2**	**3**	**2**	10
SSDD	**3**	3	7	3	**2**	3	**4**	2	**3**	**2**	**3**	3	6
COP	**3**	7	22	16	15	11	7	6	11	17	6	4	1
IMI	**3**	34	34	29	48	24	11	24	24	32	6	4	1
OS	**3**	10	19	19	13	20	**4**	10	12	19	5	4	2
PCAES	**3**	6	7	2	7	**4**	2	3	2	11	4	4	2
SV	**3**	8	19	20	8	11	**4**	10	12	17	6	4	2
Sym	**3**	**4**	18	11	13	22	6	13	**3**	24	5	4	3
ANCV	**3**	**4**	**6**	**4**	**2**	**4**	**4**	3	**3**	**2**	**3**	**2**	**11**

Definition 4. **(*within-cluster augmented non-shared nearest neighbor pair*)**

Assume that there exists any point pair x_i and x_j in cluster c_t such that $|N_S(i,j)| < \varepsilon$ and $e(x_i,x_j) \in E_{T_X}$, where ε is the threshold value. Let $\widetilde{N}_{\overline{S}}(i)_j$ and $\widetilde{N}_{\overline{S}}(j)_i$ be the ANSNN set of x_i with respect to x_j and the ANSNN set of x_j with respect to x_i, respectively. The set of within-cluster augmented non-shared nearest neighbor pairs is defined by

$$\widetilde{N}_{c_t} = \left\{ (x_u, x_v) \middle| \left(\exists x_u \in \widetilde{N}_{\overline{S}}(i)_j, \exists x_v \in \widetilde{N}_{\overline{S}}(j)_i \right) s.t.\ (x_u, x_v \in c_t) \right\} \tag{10}$$

Example 1. *Assume that the dots in Fig. 2(a) represent the data points within the same cluster c_t and the black lines represent the edges within the minimum spanning tree T_X. As an example, let us consider the point pair a and b. Here, k in the k-nearest neighbors is set to 4 for the sake of illustration. The kNN sets of a and b are $\{a_1, a_2, b, b_1\}$ and $\{b_1, b_2, b_3, a\}$, so the SNN set of a ad b is $\{b_1\}$. Furthermore, the ANSNN set of a with respect to b is $\{a, a_1, a_2\}$, while the ANSNN set of b with respect to a is $\{b, b_2, b_3\}$. In addition, the threshold value ε is set to 3 in this paper. So the point pair a and b satisfies that $|N_S(a,b)| < \varepsilon$ and $e(a,b) \in E_{T_X}$. Finally, the set of within-cluster augmented non-shared nearest neighbor pairs $\widetilde{N}_{c_t}$ is $\{(a,b),(a,b_2),(a,b_3),\cdots,(a_2,b_2),(a_2,b_3)\}$, which consists of nine point pairs.*

Definition 5. **(*within-cluster compactness*)**

If $\widetilde{N}_{c_t} \neq \varnothing$, the within-cluster compactness is defined as the average distance of all pairs of augmented non-shared nearest neighbors within cluster c_t, otherwise the within-cluster distance is determined by the

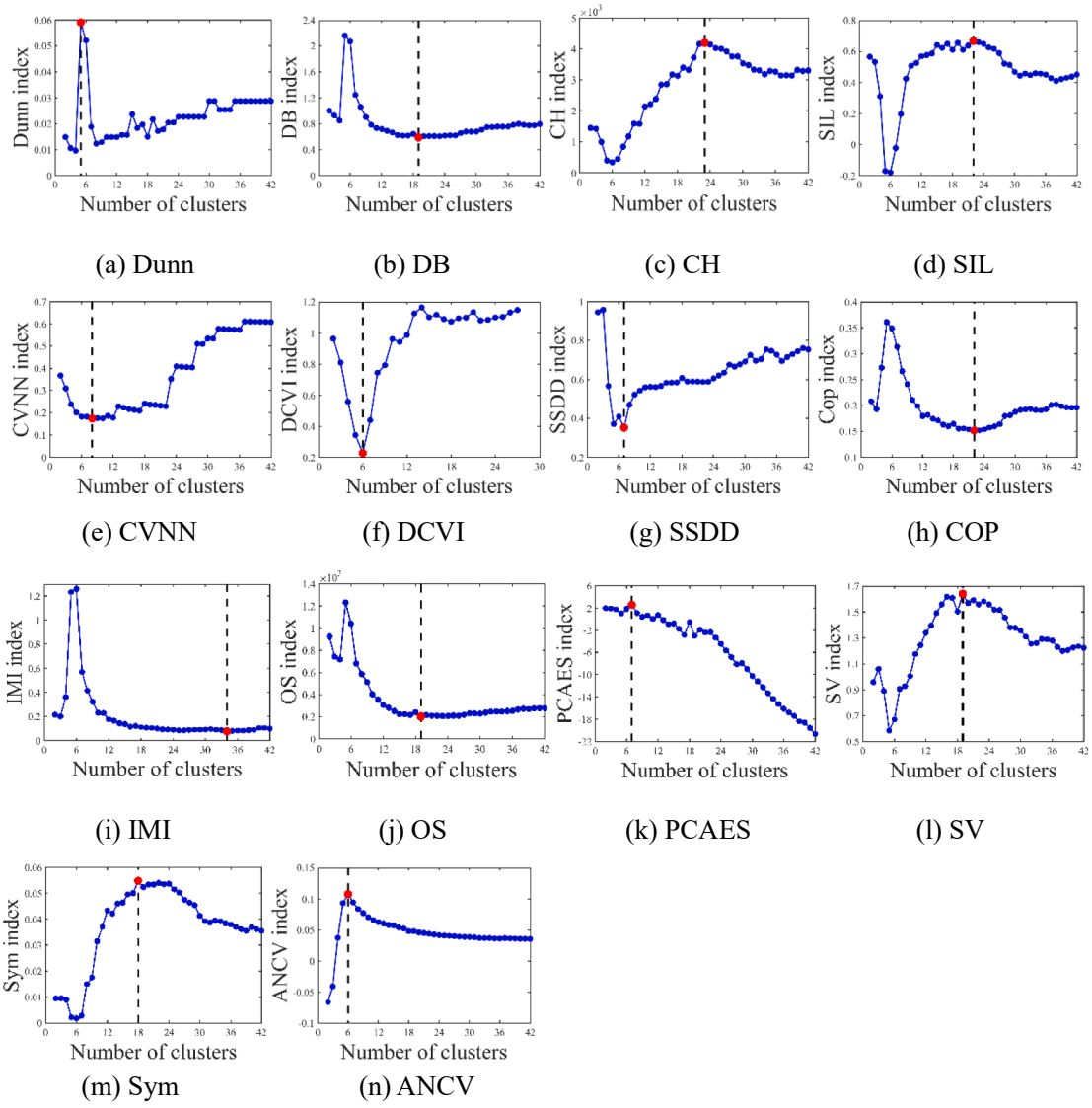


Fig. 7. The relationship between cluster number and the 14 CVI values in DS3 under the CTCEHC algorithm.

average weight of $E_{T_{c_t}}$. The within-cluster compactness of cluster c_t is defined by

$$com(c_t) = \begin{cases} \sum_{(x_u,x_v)\in\widetilde{N}_{c_t}} d(x_u,x_v) \Big/ \left|\widetilde{N}_{c_t}\right| & \text{if } \widetilde{N}_{c_t} \neq \varnothing \\ \sum_{e(x_i,x_j)\in E_{T_{c_t}}} d(x_i,x_j) \Big/ \left|E_{T_{c_t}}\right| & \text{otherwise} \end{cases} \tag{11}$$

where $E_{T_{c_t}} \subset E_{T_X}$, the edges in $E_{T_{c_t}}$ are composed of the data points in the cluster c_t, $e(x_i, x_j)$ denotes the edge associated with the data points x_i and x_j.

Definition 6. (***between-cluster augmented non-shared nearest neighbor pairs***)

Given clusters c_m and c_n, assume that there exist any points x_i and x_j such that $x_i \in c_m$,$x_j \in c_n$, and $e(x_i, x_j) \in E_{T_X}$. Let $\widetilde{N}_{\overline{S}}(i)_j$ and $\widetilde{N}_{\overline{S}}(j)_i$ be the ANSNN set of x_i with respect to x_j and the ANSNN set of x_j with respect to x_i, respectively. The set of between-cluster augmented non-shared nearest neighbor pairs is defined by

$$\widetilde{N}_{c_m,c_n} = \left\{ (x_u, x_v) \middle| \left(\exists x_u \in \widetilde{N}_{\overline{S}}(i)_j, \exists x_v \in \widetilde{N}_{\overline{S}}(j)_i \right) \; s.t. \; (x_u \in c_m, x_v \in c_n) \right\} \tag{12}$$

Example 2. *In terms of the distribution of data points,* Fig. 2(*b*) *is similar to* Fig. 2(*a*). *The difference is that* Fig. 2(*b*) *contains both clusters* c_m *and* c_n. *The data points a and b belong to* c_m *and* c_n, *respectively, and there exists* $e(a, b) \in E_{T_X}$, *then the set* $\widetilde{N}_{c_m,c_n}$ *constituted by the augmented non-shared nearest neighbor point pairs between* c_m *and* c_n *is the same as* $\widetilde{N}_{c_t}$ *in* Fig. 2(*a*), *which also includes nine point pairs.*

Definition 7. (***adjacent clusters***)

Given clusters c_m and c_n, assume that there exist any data points x_i and x_j such that $x_i \in c_m$,$x_j \in c_n$, and $e(x_i, x_j) \in E_{T_X}$, then c_m and c_n are

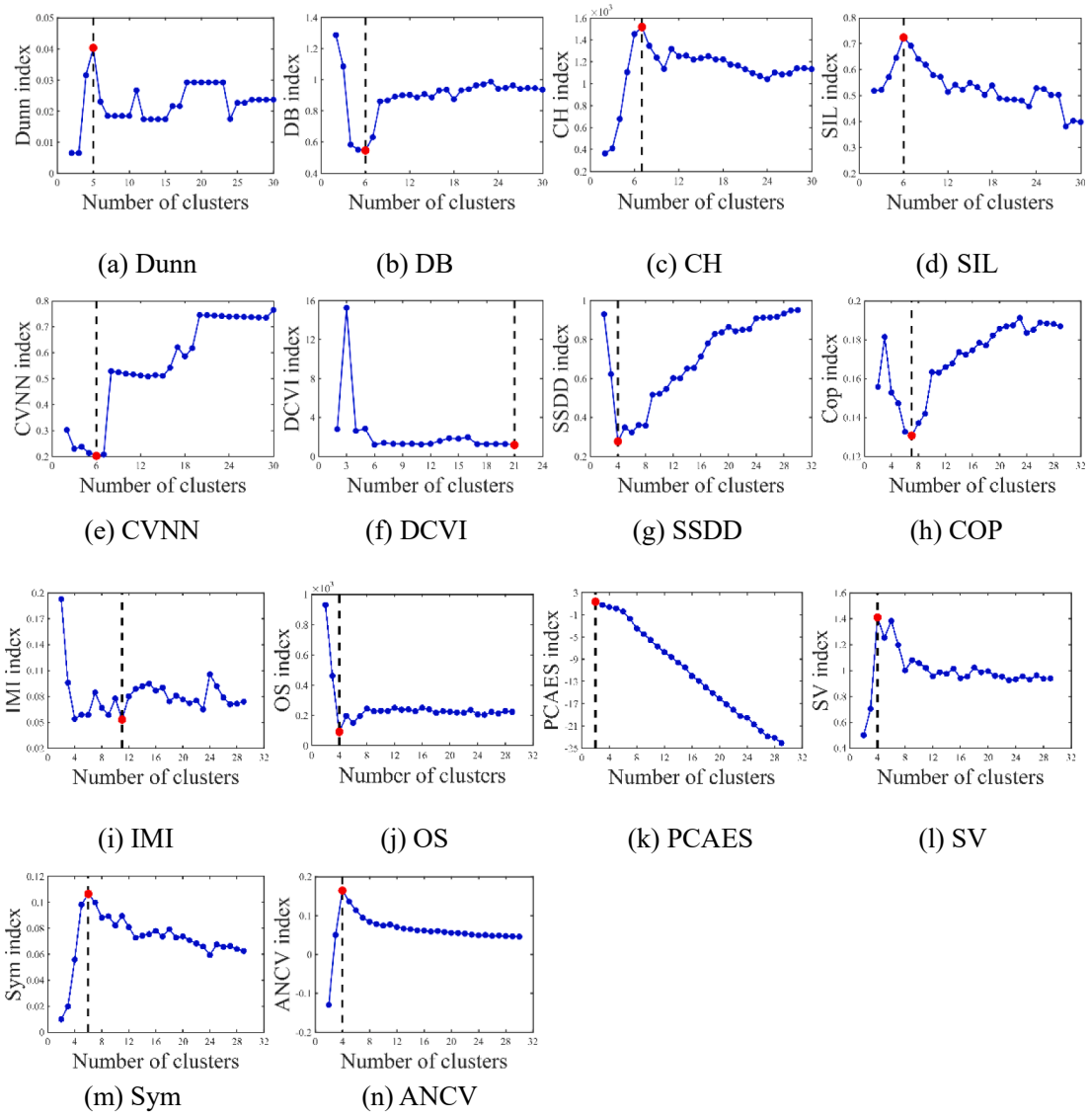


Fig. 8. The relationship between cluster number and the 14 CVI values in DS7 under the CTCEHC algorithm.

Table 5
The results on 10 real datasets.

DataSets	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	N_{hit}
T^*	3	2	2	6	3	3	2	4	2	7	
Dunn	**3**	6	5	2	**3**	9	7	**4**	4	**7**	4
DB	5	15	31	2	2	8	13	3	10	9	0
CH	5	**2**	**2**	2	**3**	5	**2**	3	**2**	2	5
SIL	**3**	6	**2**	2	**3**	5	**2**	3	**2**	**7**	**6**
CVNN	**3**	**2**	**2**	3	**3**	5	13	3	**2**	2	5
DCVI	4	**2**	6	2	2	6	**2**	26	**2**	2	3
SSDD	**3**	4	6	3	2	**3**	**2**	16	11	2	3
COP	**3**	3	4	2	2	4	**2**	**4**	**2**	10	4
IMI	**3**	13	**2**	2	2	9	10	3	7	6	2
OS	38	14	32	9	4	10	13	45	67	10	0
PCAES	2	6	14	2	**3**	8	4	**4**	**2**	2	3
SV	5	14	32	15	2	8	13	**4**	64	9	1
Sym	5	**2**	**2**	2	**3**	4	**2**	**4**	3	**7**	**6**
ANCV	**3**	15	**2**	**6**	**3**	11	13	**4**	13	**7**	**6**

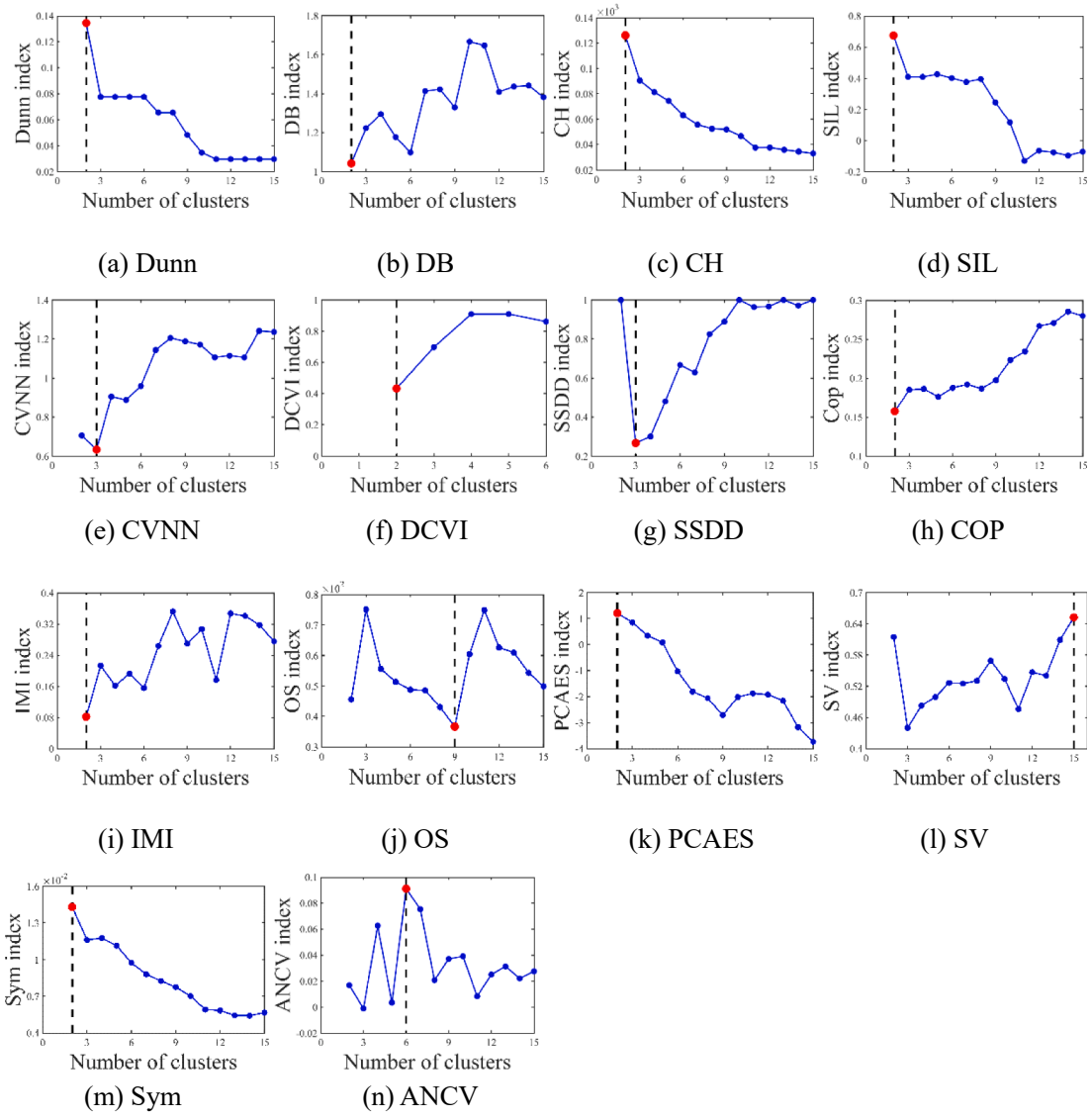


Fig. 9. The relationship between cluster number and the 14 CVI values in Glass under the CTCEHC algorithm.

adjacent clusters, denoted by $< c_m, c_n >$.

Corollary 1. *An adjacent cluster pair* $< c_m, c_n >$ *contains at least one pair of between-cluster augmented non-shared nearest neighbor pairs, i.e.,* $\widetilde{N}_{c_m,c_n} \neq \varnothing$.

Proof: ∵ The clusters c_m and c_n are adjacent

∴According to Definition 7, there exists the data point pair x_i and x_j such that $e(x_i, x_j) \in E_{Tx}$, $x_i \in c_m$,$x_j \in c_n$

∵ According to Definition 3, $x_i \in \widetilde{N}_{\overline{S}}(i)_j$,$x_j \in \widetilde{N}_{\overline{S}}(j)_i$

∴According to Definition 6, the point pair $(x_i, x_j) \in \widetilde{N}_{c_m,c_n}$.

∴An adjacent cluster pair $< c_m, c_n >$ contains at least one pair of between-cluster augmented non-shared nearest neighbor pairs, i.e., $\widetilde{N}_{c_m,c_n} \neq \varnothing$□.

Definition 8. ***(between-cluster separation)***

Given two adjacent clusters $< c_m, c_n >$, the between-cluster separation is defined as the average distance between all pairs of between-cluster augmented non-shared nearest neighbors:

$$sep(c_m, c_n) = \sum_{(x_u, x_v) \in \widetilde{N}_{cm,cn}} d(x_u, x_v) \Big/ \left|\widetilde{N}_{c_m,c_n}\right| \tag{13}$$

Definition 9. ***(ANCV index)***

Assume that the dataset X is partitioned into K clusters $C = \{c_1, c_2, \cdots, c_K\}$. The ANCV index is defined as the difference between the total between-cluster separation $SEP_K(X)$ and the total within-cluster compactness $COM_K(X)$:

$$ANCV_K(X) = SEP_K(X) - COM_K(X) \tag{14}$$

$$SEP_K(X) = \frac{1}{K-1} \sum sep(c_m, c_n) \tag{15}$$

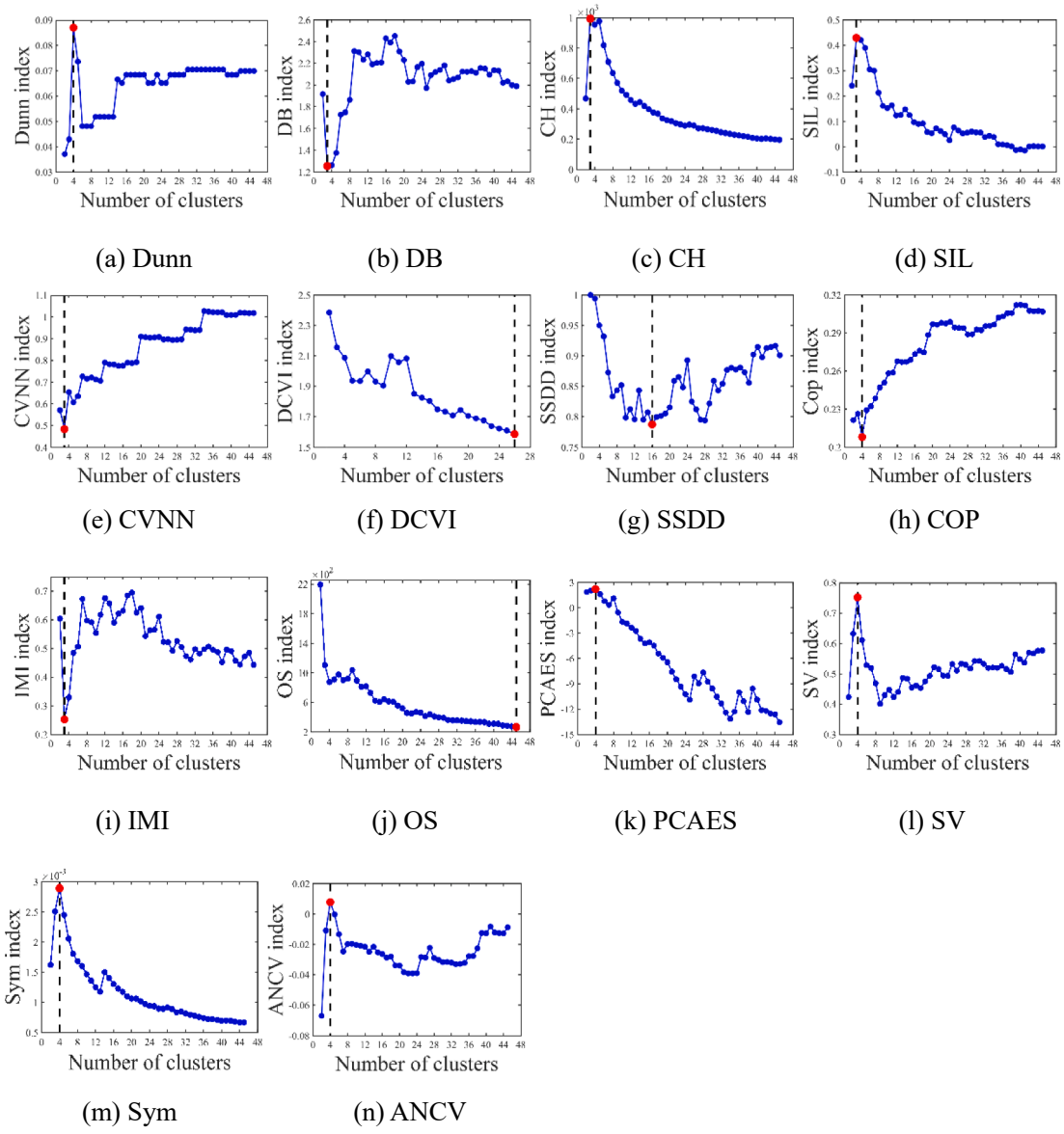


Fig. 10. The relationship between cluster number and the 14 CVI values in Wifilocalization under the CTCEHC algorithm.

Table 6
Parameter values and F1 values for Kernel k-means on 22 datasets.

Datasets	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
σ	0.20	0.12	0.12	0.12	0.15	0.16	0.23	0.14	0.15	0.42	0.17
F1	1	1	0.5481	1	0.8168	0.5743	0.8472	0.9498	1	0.9133	0.8064
Datasets	D12	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
σ	0.18	4.50	1.30	0.10	1.40	1.10	1.10	0.90	5.0	1.30	2.60
F1	0.8758	0.4143	0.5174	0.4869	0.4135	0.9714	0.4432	0.5853	0.596	0.563	0.6254

Table 7
Parameter values and F1 values for DDC on 22 datasets.

Datasets	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
k	5	5	10	10	5	10	10	10	5	10	5
λ	0.3	0.45	0.3	0.3	0.35	0.3	0.3	0.4	0.3	0.3	0.35
F1	1	1	1	1	1	1	0.9993	0.9935	1	1	1
Datasets	D12	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
k	5	5	10	5	5	10	5	5	10	5	5
λ	0.6	0.75	0.3	0.3	0.5	0.3	0.3	0.3	0.5	0.3	0.3
F1	0.991	0.4626	0.3354	0.4662	0.4758	0.4572	0.5545	0.5929	0.6809	0.5509	0.7478

Table 8
The results of five clustering algorithms on 12 two-dimensional synthetic datasets.

Datasets	T^*	Kernel k-means	CTCEHC	NTHC	SMKNN	DDC
DS1	3	**3**	**3**	**3**	**3**	**3**
DS2	4	**4**	**4**	**4**	**4**	**4**
DS3	6	20	**6**	**6**	**6**	**6**
DS4	4	**4**	**4**	**4**	**4**	**4**
DS5	2	10	**2**	**2**	**2**	**2**
DS6	4	12	**4**	**4**	**4**	**4**
DS7	4	**4**	**4**	3	3	**4**
DS8	4	13	3	**4**	2	**4**
DS9	3	**3**	**3**	**3**	**3**	**3**
DS10	2	7	**2**	**2**	**2**	**2**
DS11	3	6	**3**	**3**	**3**	**3**
DS12	2	**2**	**2**	**2**	**2**	**2**
N_{hit}		6	11	11	10	12

Table 9
The results of five clustering algorithms on 10 real datasets.

Datasets	T^*	Kernel k-means	CTCEHC	NTHC	SMKNN	DDC
R1	3	**3**	**3**	2	4	**3**
R2	2	10	15	**2**	6	15
R3	2	**2**	**2**	**2**	**2**	11
R4	6	3	**6**	11	**6**	**6**
R5	3	**3**	**3**	**3**	**3**	2
R6	3	4	11	**3**	**3**	**3**
R7	2	17	13	**2**	15	14
R8	4	**4**	**4**	3	**4**	**4**
R9	2	3	13	**2**	4	**2**
R10	7	6	**7**	**7**	5	5
N_{hit}		4	6	7	5	5

$$COM_K(X) = \frac{1}{K}\sum_{i=1}^{K} com(c_i) \tag{16}$$

Algorithm 1 gives detailed steps for implementing the ANCV index.

Algorithm 1: ANCV

Input: Data set X containing N data points, X is partitioned into 2, 3,⋯, $\sqrt{N}$ clusters $\{c_i\}$, the value of k in the k-nearest neighbors,$\epsilon = 3$
Output: Optimal cluster number Opt_K

```
1 for i = 1 to N do
2   for j = 1 to N do
3     if i ≠ j then
4       Calculate N_S(i,j),N_S̄(i)_j,Ñ_S̄(i)_j according to Eqs. (6), (7),
5       and (9), respectively
6     end
7   end
8 end
9 for K = 2 to √N do
10  for j = 1 to K do
11    Calculate Ñ_{c_j}, com(c_j), according to Eqs. (10) and (11),
12    respectively
13  end
14 end
15 for K = 2 to √N do
16  for m = 1 to K−1 do
17    for n = m+1 to K do
18      if c_m and c_n are adjacent clusters then
19        Calculate Ñ_{c_m,c_n}, sep(c_m, c_n) according to Eqs. (12) and
20        (13), respectively
21        end
22        end
23      end
24    end
25  for K = 2 to √N do
26      Calculate ANCV_K(X) according to Eqs. (14) (15) and (16)
27    end
28    Opt_K = argmax ANCV_K(X)
```

3.2. An explanation of ANCV

In this paper, we extend our analysis from the point pairs with fewer SNN within a cluster to the within-cluster ANSNN pairs, so that several small clusters are formed, and we calculate the distance between the small clusters to measure the within-cluster compactness, which has the advantage of avoiding the ANCV index being influenced by the cluster shape. For example, Fig. 3 shows the clustering results obtained using the NTHC algorithm for the Smile Face dataset (Ma, et al., 2021) by $K =$ 3, 4, and 5. As shown in Fig. 3(a), (c), and (e), the blue triangles indicate the within-cluster augmented non-shared nearest neighbor pairs, while the pairs with shared nearest neighbor number less than 3 are connected by blue lines. These point pairs form small clusters whose shape is not significantly affected by the shape of the cluster. Moreover, if the cluster number is smaller than the real number, the clustering algorithm will mistakenly merge multiple clusters into one. As an example, Fig. 3(a) shows the clustering results of three clusters, where the cluster marked with black dots indicates that the clustering algorithm wrongly merged two clusters into one. Fig. 3(a) illustrates that the distance between the augmented non-shared neighbor pairs at the intersection of the two clusters is relatively large. Since the within-cluster compactness of this cluster is large, the total within-cluster compactness $COM_3(X)$ is also of a high level. As shown in Fig. 3(c), the clustering results are correct when $K = 4$, so there is no similar situation.

When measuring the inter-cluster separation, the point pairs between adjacent clusters are selected as representative points, and the representative points are further extended to their augmented non-shared nearest neighbors so that small clusters are formed at the intersection of adjacent clusters and the inter-cluster separation can be estimated by the average distance between the point pairs between the small clusters. The proposed inter-cluster separation measure can also reduce the influence of cluster shape. In Fig. 3(b), (d), and (f), blue triangles represent the between-cluster augmented non-shared nearest neighbor pairs and blue lines represent the point pairs that have edges on the minimum spanning tree. When $K = 3$, there are between-cluster augmented non-shared nearest neighbor pairs between red and black clusters, and green and black clusters, respectively. As shown in Fig. 3 (d), when $K = 4$, the between-cluster augment non-shared nearest neighbor pairs are added between the yellow and green clusters on the basis of $K = 3$. According to Fig. 3(f), when $K = 5$, the green and yellow clusters should originally form the same cluster, and $sep(c_{red}, c_{green})$ is relatively smaller, thus making $SEP_5(X)$ smaller than $SEP_4(X)$.

Fig. 4 illustrates the clustering evaluation results of the Smile Face dataset using the ANCV index, where K varies from 2 to 26 and the red, blue, and green lines represent the within-cluster compactness, between-cluster separation and the ANCV index results of the dataset, respectively. Fig. 4 shows that $COM_4(X)$ is much smaller than $COM_2(X)$ and $COM_3(X)$ when K is equal to the true cluster number 4. The curve associated with $COM_K(X)$ tends to become more stable as K increases. Additionally, $SEP_4(X)$ is significantly larger than $SEP_3(X)$ and $SEP_5(X)$. Therefore, when $K = 4$, the ANCV index value has the highest value, thus defining 4 as the optimal cluster number.

3.3. Time complexity

In Algorithm 1, Lines 1–8 construct the minimum spanning tree T_X and calculate $N_s(i,j)$,$N_{\bar{S}}(i)_j$, and $\widetilde{N}_{\bar{S}}(i)_j$. Construction of the minimum spanning tree and computing the k-nearest neighbors of data points can be combined to reduce computing time. Given dataset X with N data points, the time complexity of constructing the minimum spanning tree using Prim algorithm (L. Liu, Ma, Zhang, Zhang, & Li, 2017) is $O(N^2)$, and the time required to calculate the shared and non-shared nearest neighbors between point pairs is about $O(N)$.

Lines 9–14 in Algorithm 1 calculates the within-cluster compactness $com(c_j)$. When data point pairs satisfying the threshold condition exist in

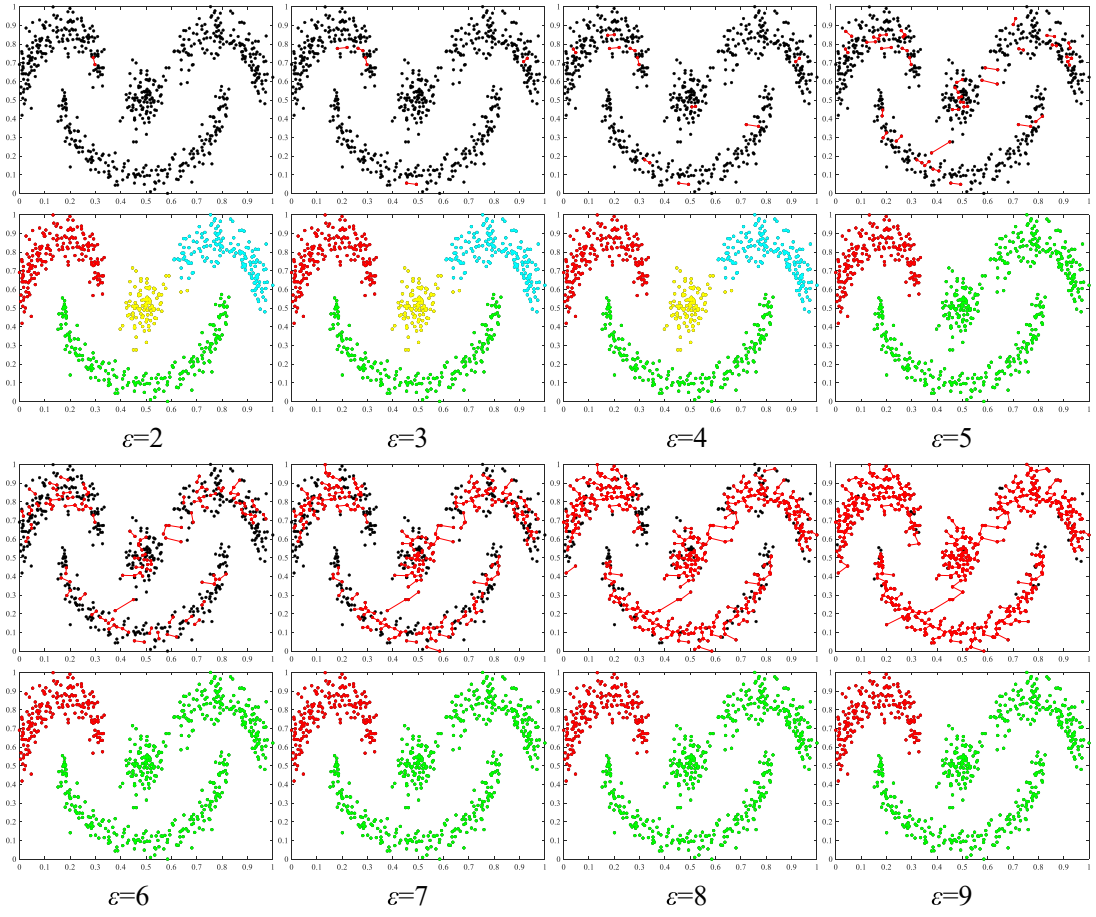


Fig. 11. The point pairs (indicated by red lines) that satisfy the number of shared nearest neighbors less than ε on the Smile Face dataset, where the clustering results of CTCEHC under the optimal number of clusters identified by ANCV are also shown.

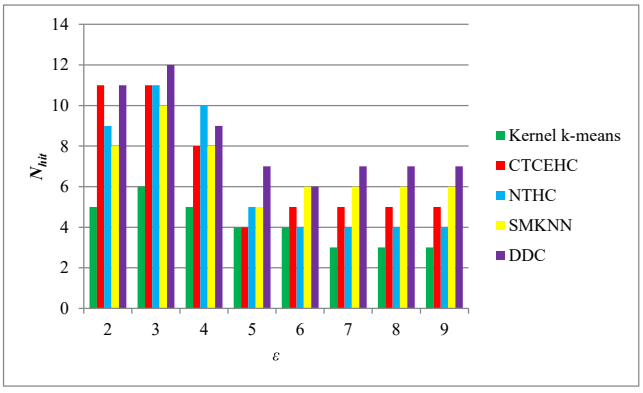


Fig. 12. The N_{hit} values for 12 two-dimensional synthetic datasets under different ε values.

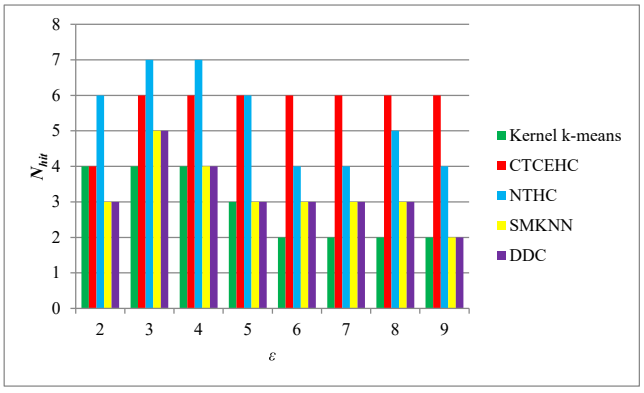


Fig. 13. The N_{hit} values for 10 real datasets under different ε values.

each cluster, the time complexity of computing the within-cluster compactness $com(c_j)$ is approximately $O(K|\widetilde{N}_{c_j}|)$. In contrast, when no data point pair satisfies the threshold condition, the time complexity of computing the within-cluster distance $com(c_j)$ is about $O(K|E_{T_{c_j}}|)$.

Lines 15–24 in Algorithm 1 calculates the between-cluster separation $sep(c_m, c_n)$. The time required to calculate the between-cluster distance $sep(c_m, c_n)$ is about $O(|\widetilde{N}_{c_m,c_n}|)$.

Lines 25–28 in Algorithm 1 calculate $ANCV_K(X)$ and the time complexity of this part can be ignored.

In summary, the time complexity required for the ANCV index is approximately $O(N^2)+O(N)+O(K|\widetilde{N}_{c_j}|)(\text{or } O(K|E_{T_{c_j}}|)) + O(|\widetilde{N}_{c_m,c_n}|)$.

4. Experiments and results

In this section, we evaluate the performance of the ANCV index using 12 two-dimensional synthetic datasets and 10 real datasets from UCI (Cheng, et al., 2018; Kwon & Sim, 2013; Xie, et al., 2020). Fig. 5 shows 12 two-dimensional synthetic datasets. Tables 1 and 2 provide the

Table 10
F1 and AMI values for CTCEHC based on the given optimal cluster number by ANCV.

Datasets	F1	AMI	Datasets	F1	AMI
DS1	1	1	**DS12**	1	1
DS2	1	1	**R1**	0.3842	0.0072
DS3	1	1	R2	0.4710	0.1761
DS4	1	1	**R3**	0.5476	0.0063
DS5	1	1	**R4**	0.6103	0.2892
DS6	1	1	**R5**	0.9571	0.8534
DS7	1	1	R6	0.4162	0.1428
DS8	0.9023	0.9157	R7	0.5197	−0.0024
DS9	1	1	**R8**	0.8719	0.7738
DS10	1	1	R9	0.3459	0.0221
DS11	1	1	**R10**	0.7691	0.8307

descriptions of the 12 two-dimensional synthetic datasets and 10 real datasets, respectively. Our experiments used five clustering algorithms, Kernel k-means, CTCEHC, NTHC, SMKNN, and DDC, to carry out the classification. A total of 13 indices are compared with the ANCV Index, including Dunn, DB, CH, SIL, CVNN, DCVI, SSDD, COP, IMI, OS, PCAES, SV, and Sym. A detailed description of these 13 indices is shown in Table 3. As a rule of thumb, we set the range of cluster numbers to $[2, \sqrt{N}]$. The experimental environment is an Intel i7-11800H computer with 16G of RAM, and the software used is MATLAB2021a. In addition, k in the k-nearest neighbors and the threshold ε in the ANCV index are set to 10 and 3, respectively. The source code of the proposed algorithm is available online at https://github.com/xjDUAN184/ANCV-validity-index/tree/master.

4.1. Test on 12 synthetic datasets

According to Table 1, DS1 is composed of three spherical clusters, while DS2 is composed of four linear clusters. DS3 and DS11 contain clusters of both linear and spherical shapes. Both DS4 and DS9 have ring-shaped and spherical clusters. Both DS5 and DS6 consist of multiple arc-shaped clusters. DS7 has four spherical clusters of varying densities. DS8 and DS12 are composed of spherical and arc-shaped clusters. DS10 consists of two ring-shaped clusters.

To evaluate the performance of the ANCV index, the 12 synthetic datasets were clustered using the CTCEHC algorithm. Fig. 6 shows the clustering results of CTCEHC. As seen in Fig. 6, CTCEHC obtained correct clustering results for all 12 synthetic datasets under the true number of clusters. And experiments were conducted on 14 validity indices, including ANCV. Table 4 lists the experimental results, where T^* represents the true cluster number, N_{hit} is the number of times the index correctly identifies the cluster number, and the bold number indicates that the result is identical to the true cluster number.

As shown in Table 4, the DB, CH, SIL, COP, and IMI indices can only correctly obtain the cluster number of DS1, but not those of the other datasets. It is primarily due to the fact that these indices are highly dependent on clustering centers, and if the clustering centers are incorrect, then the results of the indices will also be incorrect. The performance of OS and SV is slightly higher than the above-mentioned indices, and they get the correct cluster number of both DS1 and DS7 datasets. The 14 indices, except for the Dunn index, correctly identify the cluster number for DS1, which consists of three spherical clusters. The six indices Dunn, CVNN, DCVI, SSDD, Sym, and ANCV performed well on at least three datasets. The Sym index is effective for internal symmetric datasets DS1, DS2, and DS9. All four indices Dunn, DCVI, SSDD, and ANCV are effective on DS5 and DS10, which contain arc- and ring-clusters. The SSDD index can also obtain the correct cluster number for the four datasets DS1, DS7, DS9, and DS11, indicating that SSDD can identify clusters of different shapes and densities. ANCV and DCVI are both effective for at least 10 datasets, and their performance is superior to that of the other 12 indices. Overall, the ANCV index is the most stable of the 14 indices and performs well on 11 datasets.

Figs. 7 and 8 illustrate the relationship between cluster number and the 14 CVI values in DS3 and DS7 under the CTCEHC algorithm, respectively. The red dots indicate the optimal number of clusters obtained by the respective index. Since the DCVI index evaluates the clustering result according to the selected denser points, the range of the number of clusters for this index is smaller than that of the other indices. For DS3, both ANCV and DCVI indices obtained the correct optimal cluster number. DS7 consists of four spherical clusters of different densities. Despite Dunn, DB, CH, SIL, and COP performing better for spherical clusters, these indices are unable to identify the correct number of clusters due to the large difference in density between clusters. SSDD, OS, SV, and ANCV can accurately determine the optimal number of clusters for DS7.

4.2. Test on 10 real datasets from UCI

To further evaluate the performance of the ANCV index, the 10 real datasets were clustered using the CTCEHC algorithm. And experiments were conducted on 14 validity indices, including ANCV. Table 5 lists the experimental results, where T^* represents the true cluster number, N_{hit} is the number of times the index correctly identifies the cluster number, and the bold number indicates that the result is identical to the true cluster number. The datasets in Table 5 are represented by abbreviations listed in Table 2 to save table space. For the R1 dataset, the true cluster number can be detected by seven indices, including Dunn, SIL, CVNN, SSDD, COP, IMI, and ANCV. There are also seven indices valid for the R5 dataset, including Dunn, CH, SIL, CVNN, PCAES, Sym, and ANCV. Six indices can determine the correct number of clusters in each of the four datasets R3, R7, R8, and R9. Furthermore, for the R4 and R6 datasets, only one index (ANCV and SSDD, respectively) can identify the correct cluster number. For all 10 datasets, both the DB and OS indices are invalid. Both the CH and SIL indices can obtain the correct number of clusters for the R3, R5, R7, and R9 datasets. Among the 14 indices, SIL, Sym, and ANCV have the most robust performance and are valid for six datasets.

Figs. 9 and 10 illustrate the relationship between cluster number and the 14 CVI values in Glass and Wifilocalization under the CTCEHC algorithm, respectively. The red dots indicate the optimal number of clusters obtained by the respective index. For Glass, only ANCV got the correct number of clusters. Dunn, COP, PCAES, SV, Sym, and ANCV provide the correct number of clusters for Wifilocalization.

4.3. Application of ANCV to other clustering algorithms

To evaluate the performance of ANCV under different clustering algorithms, we conducted experiments on 12 synthetic and 10 real datasets using five clustering algorithms: Kernel k-means, CTCEHC, NTHC, SMKNN, and DDC. These five clustering algorithms can identify non-spherical clusters. The Kernel k-means algorithm utilizes the Gaussian kernel function, whose parameters σ are listed in Table 6. Table 7 provides values for the parameters k and λ involved in the DDC algorithm. Moreover, Tables 6 and 7 show the F1 values of Kernel k-means and DDC algorithms under the given parameters as well as the true cluster number. The larger the F1 value, the better the clustering effect. For the Kernel k-means and DDC algorithms, we used coarse grid search to determine the optimal parameter values for 22 datasets to ensure that they performed optimally with the true number of clusters. Additionally, the parameters involved in the three clustering algorithms CTCEHC, NTHC, and SMKNN are used with their default values.

In our experiments, the above five clustering algorithms were applied to the synthetic and real datasets, respectively, and then ANCV was used to obtain the optimal number of clusters. Tables 8 and 9 present the corresponding results, where T^* represents the true cluster number, N_{hit} is the number of times the index correctly identifies the cluster number, and the bold number indicates that the result is identical

to the true cluster number.

As seen in Table 8, the five clustering algorithms obtained the correct number of clusters on five datasets, DS1, DS2, DS4, DS9, and DS12. ANCV can correctly identify the number of clusters for more than 10 datasets using four clustering algorithms, CTCEHC, NTHC, SMKNN, and DDC. ANCV has the worst performance on Kernel k-means and can only identify a total of six datasets, DS1, DS2, DS4, DS7, DS9, and DS12. And all these six datasets have F1 values close to 1 in Table 6. The remaining six datasets had lower F1 values. Additionally, ANCV correctly identified all 12 datasets with DDC. According to Table 7, all 12 datasets have F1 values close to 1. Thus, ANCV can obtain the correct number of clusters for an accurate clustering result. However, when the clustering result is inaccurate, it is difficult for ANCV to determine the correct number of clusters.

We further analyze the results in Table 9. ANCV on the real dataset has a lower N_{hit} value than on the synthetic datasets. It is due to the fact that the real datasets have higher dimensionality than synthetic datasets, as well as a more complex distribution of data points. In addition, ANCV produced fewer differences in N_{hit} values for these five clustering algorithms. Similarly to the synthetic dataset, we can conclude from Tables 8 and 9 that the goodness of the clustering results influences ANCV performance.

5. Discussion

This section discusses the parameters ε involved in ANCV. According to Definition 4, only point pairs with fewer than ε shared nearest neighbors are used to calculate within-cluster compactness. Thus, within-cluster compactness is defined as the average distance of all pairs of augmented non-shared nearest neighbors within a cluster, where these pairs are derived from point pairs with a shared nearest neighbor number less than ε. ε has a value range of $[1, k+1]$, where k is the number of nearest neighbors. There are no shared nearest neighbors between point pairs when $\varepsilon = 1$, whereas all point pairs satisfy the condition when $\varepsilon = k + 1$.

Fig. 11 shows the point pairs (indicated by red lines) on the Smile Face dataset that satisfy a shared nearest neighbor number less than ε under different values of ε. In addition, Fig. 11 also shows the clustering results of CTCEHC for the optimal number of clusters identified by ANCV. Fig. 11 illustrates that when ε is small, there are fewer point pairs in the cluster, or perhaps none at all. In this case, the within-cluster compactness equals the average weight of the minimum spanning tree within the cluster according to Definition 5. This will result in a small value for within-cluster compactness, which cannot accurately reflect the distribution of data points within the cluster. As ε increases, there are more point pairs in the cluster, which indicates that most point pairs in the cluster contribute to the within-cluster compactness. As a result, the final result may not necessarily reflect the distribution of data points within the cluster. Thus, the performance of ANCV is influenced by the value of ε.

Following are the experiments conducted to determine a more appropriate ε value. We first cluster 12 two-dimensional synthetic datasets and 10 real datasets using Kernel k-means, CTCEHC, NTHC, SMKNN, and DDC five algorithms, respectively. Next, calculate the number of times (expressed by N_{hit}) the ANCV index correctly identifies the cluster number under different values of ε. Figs. 12 and 13 show the N_{hit} values of ANCV on 12 synthetic datasets and 10 real datasets, respectively. As can be seen in Figs. 12 and 13, N_{hit} values are lower when ε greater than 4. When $\varepsilon = 3$, all five clustering algorithms have larger N_{hit} values. Therefore, we set ε to 3 in this paper.

Additionally, the clustering effect plays a role in the performance of ANCV. Further explanation is provided by the following experiments. As a first step, CTCEHC is used to cluster 12 two-dimensional synthetic datasets and 10 real datasets, and then ANCV is applied to determine the optimal number of clusters. Next, we evaluate the clustering effect using F1 and AMI (Chowdhury, Bhattacharyya, & Kalita, 2021), two external clustering validity indices. Table 10 lists the values of F1 and AMI, where the bold number indicates that ANCV provides the correct number of clusters for the corresponding dataset. The greater the F1 and AMI, the better the clustering effect. Table 10 shows that, except for DS8, ANCV is effective on the remaining 11 synthetic datasets, and the corresponding F1 and AMI values reach a maximum of one. For real datasets, ANCV is effective on the R5, R8, and R10 datasets, and the corresponding F1 and AMI values on these datasets are much greater than those on the rest of the datasets. ANCV is also capable of correctly identifying the cluster number for the three datasets R1, R3, and R4 despite their small F1 and AMI values.

Because ANCV has a time complexity greater than $O(N^2)$, it is not suitable for processing massive datasets. A parallel version of ANCV can be developed to reduce its complexity. Many of the operations of ANCV are focused on constructing a minimum spanning tree. We can use a parallel algorithm on the GPU to obtain a minimum spanning tree (de Alencar Vasconcellos, Cáceres, Mongelli, & Song, 2017; Prokopenko, Sao, & Lebrun-Grandie, 2022). In addition, the calculation of within-cluster compactness focuses primarily on the determination of within-cluster augmented non-shared neighbor point pairs. Multiple threads can be assigned to different point pairs on the minimum spanning tree, which are processed by the GPU in parallel. The calculation of between-cluster separation is mainly focused on determining the augmented non-shared neighbor point pairs between clusters. Multiple threads can also be assigned to point pairs located between clusters so that the GPU can process them concurrently.

6. Conclusion

In this paper, a new cluster validity index is proposed. The proposed index is based on the point pairs with fewer shared nearest neighbors. And the within-cluster and between-cluster augmented non-shared nearest neighbors are taken as the representative points. The average distance between these representative points is taken as within-cluster compactness and between-cluster separation.

The core ideas of the proposed index include the following: (1) We search for small clusters with a relatively loose distribution within the cluster, and use the average distance between the point pairs within these small clusters as an indicator of the within-cluster compactness of the entire cluster. As a result, the index performance is less affected by the shape of the cluster. Another advantage is that when two clusters are incorrectly merged into one cluster, the within-cluster compactness of the smaller clusters within the wrongly merged cluster is greater than the within-cluster distances of the two separate clusters, respectively, thus better reflecting the distribution of data points within the cluster. (2) The average distance between pairs of data points at the intersection of two clusters is used as the between-cluster separation, making the index performance less influenced by the cluster shape. In our experiments, we selected five clustering algorithms, Kernel k-means, CTCEHC, NTHC, SMKNN, and NTHC, to cluster 12 synthetic datasets and 10 real datasets, respectively, and compared CTCEHC with Dunn, DB, CH, SIL, CVNN, DCVI, SSDD, COP, IMI, OS, PCAES, SV, and Sym for a total of 13 indices. And the experimental results showed that the ANCV index had the best performance.

As a result of our experiments, we found that it is harder to find the correct number of clusters to compute the ANCV index if the clustering results are incorrect for the real number of clusters. To address this issue, we will continue to improve the ANCV index in future studies so that it can be adapted to different clustering situations.

CRediT authorship contribution statement

Xinjie Duan: Conceptualization, Software, Validation. **Yan Ma:** Data curation, Methodology, Writing – original draft. **Yuqing Zhou:** Visualization. **Hui Huang:** Writing – review & editing. **Bin Wang:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my code in the manuscript.

Acknowledgment

This study was supported by the National Natural Science Foundation of China under Grant No. 61373004.

References

Bandyopadhyay, S., & Saha, S. (2008). A point symmetry-based clustering technique for automatic evolution of clusters. *IEEE Transactions on Knowledge and Data Engineering, 20*, 1441–1457.

Cengizler, C., & Kerem-Un, M. (2017). Evaluation of Calinski-Harabasz criterion as fitness measure for genetic algorithm based segmentation of cervical cell nuclei. *Journal of Advances in Mathematics and Computer Science, 22*, 1–13.

Cheng, D., Zhu, Q., Huang, J., Wu, Q., & Yang, L. (2018). A novel cluster validity index based on local cores. *IEEE transactions on neural networks and learning systems, 30*, 985–999.

Chowdhury, H. A., Bhattacharyya, D. K., & Kalita, J. K. (2021). UIFDBC: Effective density based clustering to find clusters of arbitrary shapes without user input. *Expert Systems with Applications, 186*, Article 115746.

de Alencar Vasconcellos, J. F., Cáceres, E. N., Mongelli, H., & Song, S. W. (2017). A parallel algorithm for minimum spanning tree on GPU. In *In 2017 International symposium on computer architecture and high performance computing workshops (SBAC-PADW)* (pp. 67–72). IEEE.

de Souto, M. C., Coelho, A. L., Faceli, K., Sakata, T. C., Bonadia, V., & Costa, I. G. (2012). A comparison of external clustering evaluation indices in the context of imbalanced data sets. In *In 2012 Brazilian Symposium on Neural Networks* (pp. 49–54). IEEE.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics, 4*, 95–104.

Gurrutxaga, I., Albisua, I., Arbelaitz, O., Martín, J. I., Muguerza, J., Pérez, J. M., & Perona, I. (2010). SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index. *Pattern recognition, 43*, 3364–3373.

Hahsler, M., Piekenbrock, M., & Doran, D. (2019). dbscan: Fast density-based clustering with R. *Journal of Statistical Software, 91*, 1–30.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193–218.

Kwon, O., & Sim, J. M. (2013). Effects of data set features on the performances of classification algorithms. *Expert Systems with Applications, 40*, 1847–1857.

Lei, Y., Bezdek, J. C., Chan, J., Vinh, N. X., Romano, S., & Bailey, J. (2016). Extending information-theoretic validity indices for fuzzy clustering. *IEEE Transactions on Fuzzy Systems, 25*, 1013–1018.

Li, R., & Cai, Z. (2022). A clustering algorithm based on density decreased chain for data with arbitrary shapes and densities. *Applied Intelligence*, 1–12.

Liang, S., Han, D., & Yang, Y. (2020). Cluster validity index for irregular clustering results. *Applied Soft Computing, 95*, Article 106583.

Liu, L., Ma, Y., Zhang, X., Zhang, Y., & Li, S. (2017). High discriminative SIFT feature and feature pair selection to improve the bag of visual words model. *IET Image Processing, 11*, 994–1001.

Liu, X. (2022). Simplemkkm: Simple multiple kernel k-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, X., Zhu, X., Li, M., Wang, L., Zhu, E., Liu, T., … Gao, W. (2019). Multiple kernel k-means with incomplete kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*, 1191–1204.

Liu, Y., Jiang, Y., Hou, T., & Liu, F. (2021). A new robust fuzzy clustering validity index for imbalanced data sets. *Information Sciences, 547*, 579–591.

Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J., & Wu, S. (2013). Understanding and enhancement of internal clustering validation measures. *IEEE transactions on cybernetics, 43*, 982–994.

Ma, Y., Lin, H., Wang, Y., Huang, H., & He, X. (2021). A multi-stage hierarchical clustering algorithm based on centroid of tree and cut edge constraint. *Information Sciences, 557*, 194–219.

Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of multivariate analysis, 98*, 873–895.

Pakhira, M. K., Bandyopadhyay, S., & Maulik, U. (2004). Validity index for crisp and fuzzy clusters. *Pattern recognition, 37*, 487–501.

Pfeifer, B., & Schimek, M. G. (2021). A hierarchical clustering and data fusion approach for disease subtype discovery. *Journal of Biomedical Informatics, 113*, Article 103636.

Pfitzner, D., Leibbrandt, R., & Powers, D. (2009). Characterization and evaluation of similarity measures for pairs of clusterings. *Knowledge and Information Systems, 19*, 361–394.

Prokopenko, A., Sao, P., & Lebrun-Grandie, D. (2022). A single-tree algorithm to compute the Euclidean minimum spanning tree on GPUs. In *Proceedings of the 51st International Conference on Parallel Processing* (pp. 1-10).

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association, 66*, 846–850.

Rendón, E., Abundez, I., Arizmendi, A., & Quiroz, E. M. (2011). Internal versus external cluster validation indexes. *International Journal of computers and communications, 5*, 27–34.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science, 344*, 1492–1496.

Rojas-Thomas, J., Santos, M., & Mora, M. (2017). New internal index for clustering validation based on graphs. *Expert Systems with Applications, 86*, 334–349.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*, 53–65.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal, 27*, 379–423.

Singh, A. K., Mittal, S., Malhotra, P., & Srivastava, Y. V. (2020). Clustering Evaluation by Davies-Bouldin Index (DBI) in Cereal data using K-Means. In *In 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 306–310). IEEE.

Starczewski, A. (2017). A new validity index for crisp clusters. *Pattern Analysis and Applications, 20*, 687–700.

van der Hoef, H., & Warrens, M. J. (2019). Understanding information theoretic measures for comparing clusterings. *Behaviormetrika, 46*, 353–370.

Wang, Y., Ma, Y., & Huang, H. (2021). A neighborhood-based three-stage hierarchical clustering algorithm. *Multimedia Tools and Applications, 80*, 32379–32407.

Wang, Y., Ma, Y., Huang, H., Wang, B., & Acharjya, D. P. (2023). A split–merge clustering algorithm based on the k-nearest neighbor graph. *Information Systems, 111*, Article 102124.

Wu, K.-L., & Yang, M.-S. (2005). A cluster validity index for fuzzy clustering. *Pattern Recognition Letters, 26*, 1275–1291.

Xie, J., Xiong, Z.-Y., Dai, Q.-Z., Wang, X.-X., & Zhang, Y.-F. (2020). A new internal index based on density core for clustering validation. *Information Sciences, 506*, 346–365.

Yang, J., Ma, Y., Zhang, X., Li, S., & Zhang, Y. (2017). An initialization method based on hybrid distance for k-means algorithm. *Neural computation, 29*, 3094–3117.

Žalik, K. R., & Žalik, B. (2011). Validity index for clusters of different sizes and densities. *Pattern Recognition Letters, 32*, 221–234.

Zhao, Q., & Fränti, P. (2013). Centroid ratio for a pairwise random swap clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering, 26*, 1090–1101.

Zhou, S., Liu, F., & Song, W. (2021). Estimating the Optimal Number of Clusters Via Internal Validity Index. *Neural Processing Letters, 53*, 1013–1034.