Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 2959017, 10 pages
https://doi.org/10.1155/2019/2959017

Research Article

A Novel Local Density Hierarchical Clustering Algorithm Based on Reverse Nearest Neighbors

Yaohui Liu,[1,2] Dong Liu,[2] Fang Yu,[2] and Zhengming Ma[1]

[1]*School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, Guangdong 510006, China*
[2]*School of Software and Communication Engineering, Xiangnan University, Chenzhou, Hunan 423000, China*

Correspondence should be addressed to Fang Yu; yfjammy@163.com

Received 7 May 2019; Revised 6 July 2019; Accepted 24 July 2019; Published 5 August 2019

Academic Editor: Wanquan Liu

Clustering is widely used in data analysis, and density-based methods are developed rapidly in the recent 10 years. Although the state-of-art density peak clustering algorithms are efficient and can detect arbitrary shape clusters, they are nonsphere type of centroid-based methods essentially. In this paper, a novel local density hierarchical clustering algorithm based on reverse nearest neighbors, RNN-LDH, is proposed. By constructing and using a reverse nearest neighbor graph, the extended core regions are found out as initial clusters. Then, a new local density metric is defined to calculate the density of each object; meanwhile, the density hierarchical relationships among the objects are built according to their densities and neighbor relations. Finally, each unclustered object is classified to one of the initial clusters or noise. Results of experiments on synthetic and real data sets show that RNN-LDH outperforms the current clustering methods based on density peak or reverse nearest neighbors.

1. Introduction

Clustering is the task to find a set of groups in which similar objects are in the same group, but different objects are separated into different groups. Since clustering can uncover the inherent, potential, and unknown knowledge, principles, or rules in the real-world, it has been widely used in many fields, including data mining, pattern recognition, machine learning, information retrieval, image analysis, and computer graphics [1–3]. According to the strategies used, clustering algorithms are traditionally classified into connectivity-based approaches, centroid-based approaches, distribution-based approaches, and density-based approaches [1, 2]. Among these kinds of approaches, density-based approaches allow to discover clusters with arbitrary shapes and different sizes without specifying the number of clusters.

In density-based clustering, clusters are considered to be dense regions of objects separated by low-density regions representing noise. With respect to clustering, the procedure can be broken up into two steps: estimating the density of each object and grouping density-connected objects.

The first approach adopted the density-based strategy proposed by Ester et al. [4] in the paper "Density-Based Spatial Clustering of Applications with Noise," which is dubbed as DBSCAN. In this approach, the density of each object is defined as the number of objects contained in its eps neighborhood. If the number is greater than minpts, the object is regarded as core objects, otherwise, as noise. Then, all objects that are reachable from one of the unclustered core objects are grouping to a cluster. However, it is difficult for DBSCAN to select two proper parameters eps and minpts. Another drawback of DBSCAN is that the adjacent clusters of different densities could not be properly identified [5, 6].

Density peak clustering (DPC) [7] is another famous strategy for density-based clustering, which is based on the idea that cluster centres have higher densities than their neighbors and are far away from each other. This method can identify the cluster centres from the decision graph, which is constructed by the density and the distance attributes of each objects. Moreover, it only needs one parameter. Although it seems more convenient than DBSCAN in completing clustering, it has some inner defaults. First, it is a nonsphere type of centroid-based method essentially according to DPC's

definition of density peak and the strategy of grouping. So, in some cases, complex shapes still cannot be recognized by this method. Second, the cluster centres are picked out from the decision graph manually, which limits the application of DPC. Besides, it is very difficult to select the true centres on some specific data sets. Third, errors will be propagated in the subsequent assignment process.

To remedy these limitations in DPC, there are many improved methods that have been proposed [5, 8–14]. FKNN-DPC [8] defines a uniform local density metric based on the k-nearest neighbors and uses a fuzzy technique to complete the assignment procedure after the cluster centres have been found out manually. ADPC [5] calculates local density of each object on its k-nearest neighbors by using Gaussian kernel function and applies the divide-and-conquer strategy to find cluster centres and group other objects automatically. RECOME [10] defines a new density measure as the ratio of each object's density to the maximum density of its k-nearest neighbors and also uses the divide-and-conquer strategy to partition a data set. Although these algorithms have improved DPC in some aspects, they still suffer from some drawbacks of centroid-based methods.

In contrast to the algorithms listed above, RECORD [15], RNN-DSC, [16], IS-DSC [17], and ISB-DSC [6] use reverse nearest neighbors to define object density. From the graph theory angle to interpret, these algorithms use the directed graph to complete clustering. In the graph, each vertex is an object of a data set, and for any two vertexes a and b, there is a directed edge from a to b if b is one of the k-nearest neighbors of a. RECORD defines those vertexes as core objects whose out degrees are not lower than the input parameter k and outliers otherwise. Outliers are regarded as noises and eliminated from the graph, while core vertexes and their edges form a subgraph, from which all strong connected components are found out as the result of clustering. The main distinction between RNN-DSC and RECORD lies on the outliers' assignment. In the former, an outlier will be grouped into the cluster that its nearest neighbor belongs to if the nearest neighbor is a core object. IS-DSC defines the k-influence space of each object as the intersection of its k-nearest neighbors set and reverse k-nearest neighbors set. The method applies the STRATIFY algorithm to remove outliers firstly and then performs the similar clustering procedure on remaining objects as RECORD, but each vertex is defined as the core object if its size of k-influence space is greater than 2k. ISB-DSC also uses k-influence space to create subgraph like IS-DSC, but the clustering procedure is applied on the whole data set. Comparing to DPC, the superiority of these approaches is that they no longer need to find the cluster centres. However, since RECORD, IS-DSC, and ISB-DSC employ a global threshold to predetermine outliers, they partition too many objects to noise. PIDC [18] uses the size of the unique closest neighbor set as an estimate of object density and growing strategies to complete clustering. Although this method is parameter independent, it is sensitive to noise and has high computing complexity.

In this paper, we propose an improved clustering approach by combining the k-reverse nearest neighbor graph model and density hierarchical relationship model. Based on the reverse nearest neighbor model and parameter k, a directed graph is constructed from objects of a data set. By searching strong connected components in the graph, the data set is partitioned into several initial clusters. Then, we use density dependence and k nearest neighborhood to build the density hierarchical relationships of all objects. Each unclassified object is grouped into the same cluster in which its parent is. The algorithm has the following advantages: (1) The method searches core regions instead of density peaks. So it can find out the true clusters automatically rather than to get some false cluster centres. (2) A novel density measure is proposed based on the k reverse nearest neighbor, which can reflect the aggregate relations of objects. (3) It is more efficient that our approach classifies the unclustered objects by the local density hierarchy relationships. It also reduces the risk of misclassification by the orders of the objects.

The proposed algorithm is performed on synthetic and real-world data sets, which are widely used for the performance tests of clustering algorithms. The results of RNN-DPC are compared with IS-DSC, ISB-DSC, RNN-DSC, and ADPC in terms of three very popular benchmarks: F-measure (F1) [19], adjusted mutual information (AMI), and Adjusted Rand Index (ARI) [20]. The ratio of the noise number to total objects number is taken as the benchmark too.

The rest of the paper is organized as follows: Section 2 makes a detailed description of the notations and definitions used in our algorithm. Section 3 describes the procedures of RNN-LDH in detail. Section 4 gives our experiment results and discusses the choice of parameter k briefly. Section 5 draws some conclusions.

2. RNN-DHR Algorithm

In this section, we give the detail description of RNN-LDH theoretically. Some definitions in the section were introduced in other papers but modified by our method.

2.1. Notations and Definitions. The notations used in this paper are listed below:

(1) $|\cdot|$: cardinal of a set

(2) X: set of data with d dimension; d: the dimension number of data

(3) x, y, z: any three objects in X

(4) $d(x, y)$: distance between two objects x and y

(5) $N_k(x)$: set of the k-nearest neighbor of object x; k: the input parameter with the integer value

(6) Especially, $N_1(x)$ is the set containing only one object which is nearest to object x

(7) $R_k(x)$: set of the k-reverse nearest neighbor of object x, which is defined as

$$R_k(x) = \{y \mid y \in X, x \in N_k(y)\}. \quad (1)$$

(8) std: standard deviation function.

Definition 1 (directly density reachable). Object x is directly density reachable from an object y if

(1) $|R_k(x)| \geq k$ and $|R_k(y)| \geq k$

(2) $x \in N_k(y)$ and $y \in N_k(x)$

Definition 2 (density reachable). Object x is directly density reachable from object y if there is a chain of objects $x_1, \ldots, x_m$, $x = x_1$, and $y = x_m$, which satisfies the conditions listed below:

(1) $\forall i, 1 \leq i \leq m$, x_i is a core object

(2) $\forall i, 1 \leq i < m$, x_i is directly density reachable from x_{i+1}

Definition 3 (core region). A core region (R_c) is a none empty subset of X such that

(1) $|R_c| > k/d$

(2) $\forall x, y \in R_c$, x is density-reachable from y

Definition 4 (extended core region). Given a core region (R_c), its extended core region (R_e) composes of all elements in R_c and any object x which is satisfying the following conditions:

(1) x does not belong to any core region

(2) $\exists y \in R_c$, $x \in R_k(y)$ and $N_1(x) = \{y\}$

Definition 5 (local density). Local density of an object x is defined as

$$\rho(x) = \sum_{y \in R_k(x)} |R_k(y)| \cdot e^{\left(-d^2(x,y)/\delta^2\right)}, \tag{2}$$

where $\delta = \max_{x \in X} (\min_{y \in R_k(x)} d(x, y))$.

Definition 6 (parent). The parent of an object x is defined as

$$\text{parent}(x) = \begin{cases} y, & \text{if } H \neq \varnothing, \\ x, & \text{otherwise,} \end{cases} \tag{3}$$

where $H = \{y \mid y \in N_k(x), \rho(y) \geq \rho(x), \text{ and } d(x, y) \leq \delta\}$.

The parent represents a local density hierarchical relationship of object x to its k-nearest neighbors.

Definition 7 (hierarchical distance). The hierarchical distance of an object x is defined as

$$d_h(x) = \begin{cases} d(x, y), & \text{if parent}(x) = y, \\ \min\limits_{y \in N_k(x)} d(x, y), & \text{otherwise.} \end{cases} \tag{4}$$

Definition 8 (inner distance). The inner distance of an extended core region R_e is defined as

$$d_c = \max_{x \in R_e} d_h + b, \tag{5}$$

where $b = \text{std}\{d(x, y) \mid x \in X, y = N_1(x)\}$ is the standard deviation of distances of all objects to their closest neighbors. Usually, the distance of a core object to its closest core object is less than the distance to its closest boundary object. The offset b can help d_c to capture the global distribution of objects.

Definition 9 (density connected). An object x is density connected to an extended core region R_e if there exists an object u and a chain $x_1, \ldots, x_m$, $x = x_1$, and $y = x_m$ such that

(1) $y \in R_e$

(2) $\text{parent}(x_i) = x_{i+1}$ and $d(x_i, x_{i+1}) \leq d_c$

Definition 10 (cluster). Given an extended core region R_e, a cluster C is the union of R_e and all objects x in X which are density connected to the R_e.

Definition 11 (noise). An object x is a noise if it does not belong to any cluster of X.

NR is the noise ratio, which is defined as NR = |noise|/|X|.

3. Procedures of the RNN-DHR Algorithm

In this section, we discuss our algorithm in detail.

Algorithm 1 lists the procedures for performing RNN-LDH, which accepts two inputs: data set X and nearest neighbor parameter k and outputs a label vector. The value of each element in the label vector indicates which cluster that the corresponding object belongs to, and the object is a noise if its label value is zero.

In the procedure of Algorithm 2, function *GetLDH* is called to get the local density hierarchical relationship (parent) of each object and the result is saved into the array variant *parent* firstly; then, from step 5 to 18, all extended core regions in data set X are found out by calling the *FindECR* procedure and saved to set variant *ECRs*, and each extended core region is an initial cluster; finally, from steps 20 to 29, each noise object connected to an initial cluster is identified as the same cluster by an iterative way if it satisfies the distance condition.

The algorithm *GetLDH* is realized according to Definition 6. The main purpose of this algorithm is to find out each object's density-dependent object dubbed as *parent* in its k-nearest neighbors. Meanwhile, the hierarchical distance of each object is calculated by formula (5) and saved into d_h array.

Figure 1 shows the result of *GetLDH* algorithm on Compound [21] data set. In this figure, the red circle represents the core object and the black circle represents the boundary or noise. The larger the density of the object is, the bigger the circle shows. A line with direct arrow represents

```
(1) label(∀x ∈ X) ⟵ UNLABELED;
(2) parent ⟵ GetLDH(X, k);
(3) cid ⟵ 1;
(4) ECRs ⟵ {};
(5) for all x ∈ X
(6)    if label(x) = UNLABELED
(7)       if |R_k(x)| ≥ k
(8)          R_e ⟵ FindECR(x, k, cid);
(9)          if R_e ≠ ∅
(10)            d_c(cid) ⟵ max_{y∈R_e} d_h(y) + b;
(11)            ECRs{cid} ⟵ R_e;
(12)            cid ⟵ cid + 1;
(13)         end if
(14)      else
(15)         label(x) ⟵ NOISE;
(16)      end if
(17)   end if
(18) end for
(19) bChanged ⟵ TRUE;
(20) while (bChanged)
(21)    bChanged ⟵ FALSE;
(22)    for all label(x) = NOISE
(23)       cid ⟵ label(parent(x));
(24)       if cid ≠ Noise && d(x, parent(x)) ≤ d_c(cid)
(25)          label(x) ⟵ cid;
(26)          bChanged ⟵ TRUE;
(27)       end if
(28)    end for
(29) end while
(30) return label;
```

Algorithm 1: *RNN-LDH(X, k)*.

```
(1) for all x ∈ X
(2)    d_min ⟵ δ;
(3)    y ⟵ x;
(4)    for each o ∈ N_k(x)
(5)       if parent(o) ≠ x&ρ(o) ≥ ρ(x)&d(x, o) ≤ d_min
(6)          d_min ⟵ d(x, o);
(7)          y ⟵ o;
(8)       end if
(9)    end for
(10)   if y ≠ x
(11)      parent(x) = y;
(12)      d_h(x) = d(x, y);
(13)   else
(14)      parent(x) = x;
(15)      d_h(x) = min_{y∈N_k(x)} d(x, y);
(16)   end if
(17) end for
(18) return parent;
```

Algorithm 2: *GetLDH(X, k)*.

the local density hierarchical relationship of two connected objects.

Algorithm 3 represents the processing of *FindECR* for finding a new cluster. An unlabelled core object x is input as a seed and appended into a queue. The algorithm pops

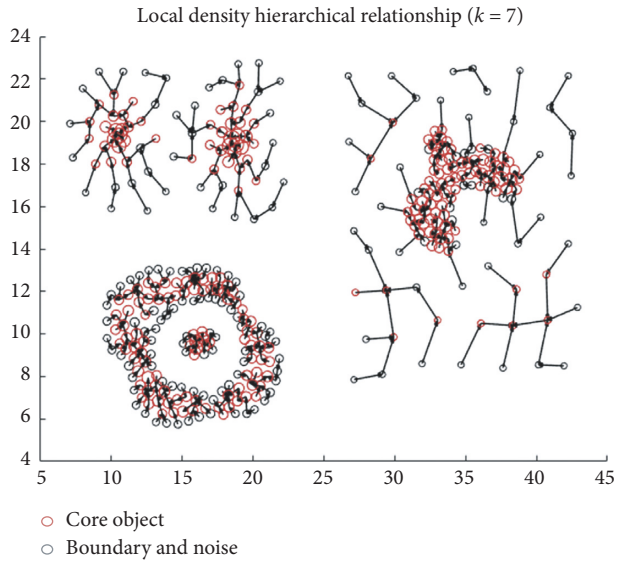

Figure 1: Local density hierarchical relationship (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

```
(1)  R_e ⟵ {};
(2)  initialize an empty queue Q;
(3)  Q.enqueue(x);
(4)  while not empty Q
(5)  y ⟵ Q.dequeue();
(6)  label(y) ⟵ cid;
(7)  R_e ⟵ R_e ∪ {y};
(8)  for each o ∈ N_k(y)
(9)     if label(o) = UNLABELED & y ∈ N_k(o)
(10)       if |N_k(o)| ≥ k
(11)          Q.enqueue(o);
(12)       else
(13)          label(o) ⟵ NOISE;
(14)       end if
(15)    end if
(16) end for
(17) end while
(18) if |R_e| ≤ k/d
(19)    label(∀o ∈ R_e) ⟵ NOISE;
(20)    return {};
(21) end if
(22) for each y ∈ R_e
(23)    for each o ∈ N_k(y)
(24)       if y ∈ N_1(o) & label(o) = UNLABELED
(25)          R_e ⟵ R_e ∪ {o};
(26)          label(o) ⟵ cid;
(27)       end if
(28)    end for
(29) end for
(30) return R_e;
```

Algorithm 3: *FindECR(x, k, cid)*.

the first seed of the queue and performs the searching procedure in k-nearest neighbors of the seed iteratively. For all unlabelled objects which are visited in the

procedure, those core objects are set to the same cluster number *cid*, while the others are labeled as noise. Step 3 to Step 17 finds out a core region starting from core object x. Steps 18–21 discard this core region if its size is not greater than k/d. Step 22–29 extends the core region by using Definition 4.

Figure 2 shows the result of the *FindECR* algorithm on Compound [21] data set. In this figure, six extended core regions are found out and black dots represent the boundary and noise. The final results of our algorithm and the comparison with the state-of-art methods on Compound will be shown in the next section.

3.1. Choice of k. The five algorithms discussed in this paper all need one parameter k. IS-DSC and ADPC did not give the way how to set the value of k. ISB-DSC compared its parameter setting with DBSCAN and drew a conclusion that it is more robust than DBSCAN for different setting of k, but it did not address the choice of k too. RNN-DSC discussed 2 approaches to determine an appropriate value of k. Each approach chose the best k from 1 to 100 by a criterion. RNN-LDH also can use these two ways to choose k. By analysing results of large amount of experiments, we cannot yet find out the theoretical bases of k choice. For achieving the best performance of each testing algorithm, we choose the k in the range independently. By this value, the number of clusters grouped by the algorithm is as close as possible to the true class number and F1 measure is largest.

3.2. Complexity of the Algorithm. The time complexity of RNN-LDH depends on the following aspects: (1) computing the distance between points $O(n^2)$; (2) sorting the distance vector of each object ($O(n^2)$), the time complexity will be down to $O(n\log(n))$; (3) computing the local density ρ with k-reverse nearest neighbors ($O(kn)$) but k is not great than n; (4) calculating the distance d_h for each object ($O(kn)$); (5) finding extended core regions ($O(n^2)$); and (6) classifying noise ($O(n^2)$). So the overall time complexity of RNN-LDH is $O(n^2)$.

The above analysis shows that RNN-LDH has the same complexity as RNN-DSC and ADPC.

4. Results and Discussion

To evaluate the performance of RNN-LDH, we perform a set of experiments on synthetic and real world data sets which are commonly used to test the performance of clustering algorithms. Indeed, we compare the performance of RNN-LDH with well-known clustering algorithms including RNN-DSC in [15], IS-DSC in [16], ISB-DSC in [6], and ADPC in [5]. Three popular criteria F1 measure (F1) [19], adjusted mutual information (AMI), and adjusted rand index (ARI) [20] are used to evaluate the performance of the above clustering algorithms. The upper bounds of these criteria are all 1.0. The better the clustering is, the larger the benchmark values are.

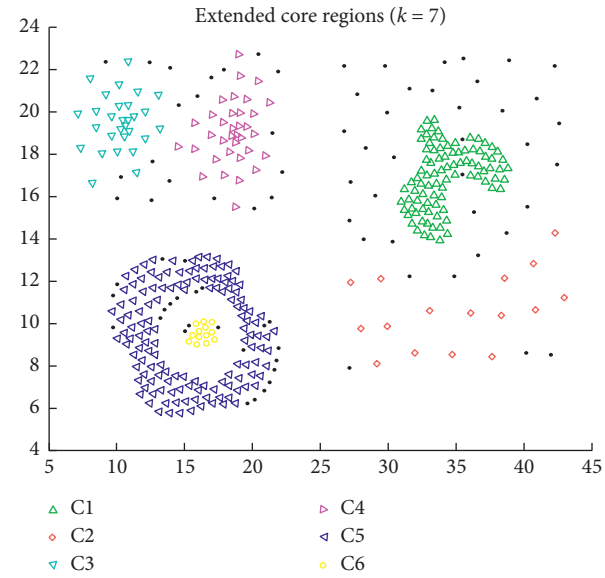

Figure 2: Extended core regions (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

4.1. Synthetic Data. Table 1 shows the synthetic data sets we used in this paper. These data sets are all composed of classes with different densities, shapes, and orientations. The first 6 data sets were obtained from [21], and the remains were downloaded from [22]. The result of each algorithm for some of these synthetic data sets is displayed in Figure 3, plotted by different marks and color points, and all noises are plotted as black points. The parameter setting (k), cluster number found (C), noise ratio (NR), and values of benchmarks as F1, AMI, and ARI are listed in Table 2.

There are 300 objects in path-based data set. They are classified to 3 classes. One class forms a 3/4 circular ring, and the other two classes distribute at the both ends of the horizontal diameter of the ring. As shown in the first row of Figure 3, RNN-LDH gets the best result, RNN-DSC also gets the correct number of clusters, and the other three algorithms classify the data set incorrectly.

Compound has six classes with different densities. Two adjacent classes in the upper-left corner are subject to Gaussian distribution, and in the right of the figure, the class with the irregular shape is surrounded by the class with lowest density. In the bottom-left corner, the smallest class is encircled by the ring-shape class. As shown in the second row of Figure 3, our method partitions three classes exactly which are labeled as yellow hexagram, blue left-pointing triangle, and fuchsia upward-pointing triangle, and one object in the contiguous zone of two classes in the upper-left corner is classified incorrectly. Only part objects (green diamond) in the lowest density class are recognized, and unrecognized objects are labeled as noise (black points). Although all objects are classified to one of six classes by RNN-DSC, many objects are partitioned wrong. Although IS-DSC gets the best benchmarks, it only finds out core objects, but too many other objects are treated as noise. ISB-DSC and ADPC even cannot find correct number of classes.

Table 1: Synthetic data sets.

Data	Objects	Dimensions	Classes
Pathbased [20]	300	2	3
Compound [20]	399	2	6
Flame [20]	240	2	2
Dim1024 [20]	1024	1024	16
Spiral [20]	312	2	3
Jain [20]	373	2	2
t4.8k [21]	8000	2	6
t5.8k [21]	8000	2	6
t7.10k [21]	10000	2	9
t8.8k [21]	8000	2	8

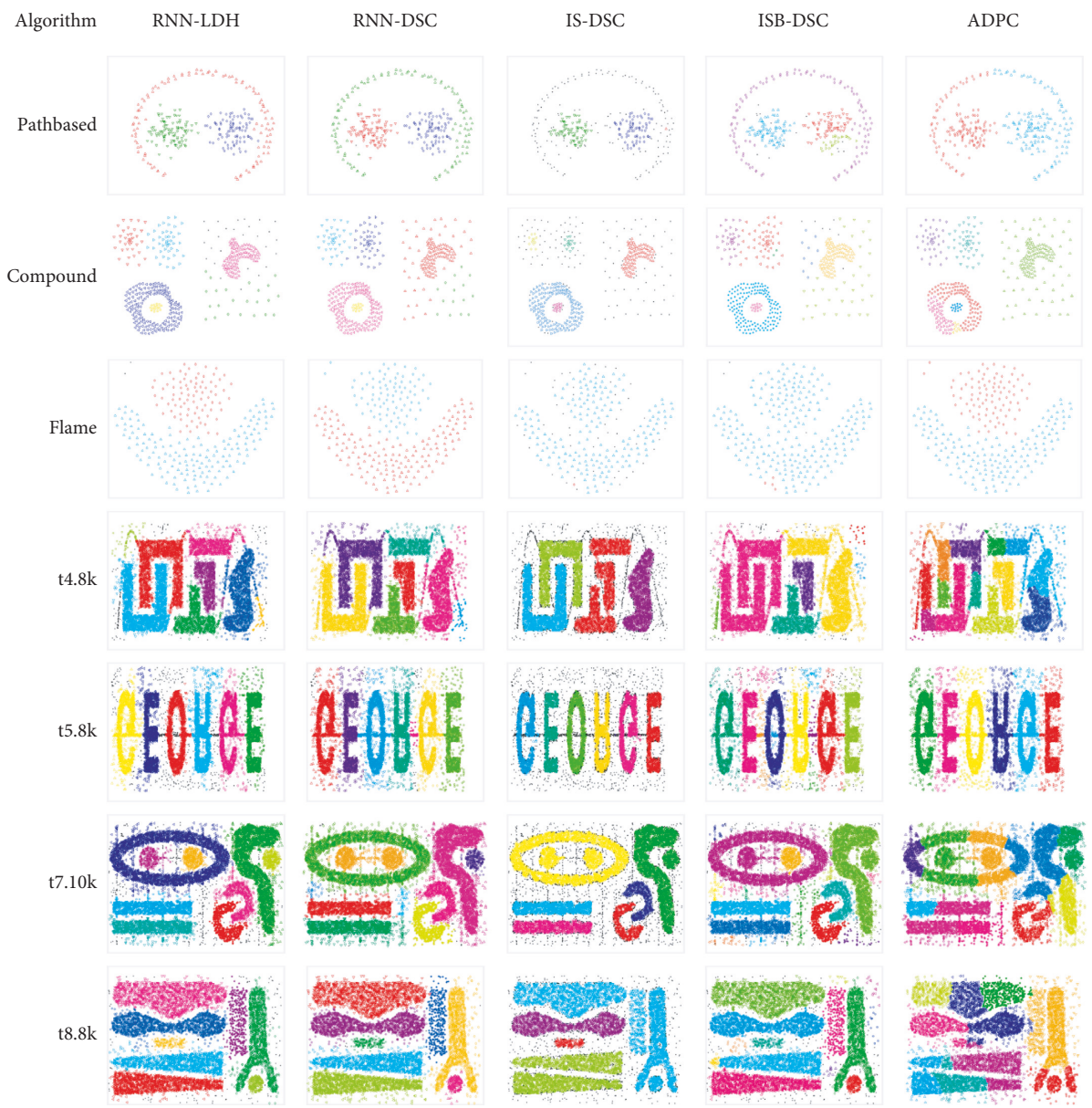

Figure 3: Comparison of RNN-LDH with RNN-DSC, IS-DSC, ISB-DSC, and ADPC. Different clusters are marked by different markers and colors (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Table 2: Results of the algorithm on synthetic data sets.

Algorithms	k	C	F1	AMI	ARI	NR (%)	k	C	F1	AMI	ARI	NR (%)
				Pathbased						cluto-t4.8k		
RNN_LDH	6	3	**1**	**1**	**1**	**0.00**	34	*8*	0.91	0.87	0.91	1.70
RNN_DSC	6	3	0.99	0.94	0.96	0.30	31	*7*	0.88	0.85	0.87	0.10
IS_DSC	10	3	**1**	**1**	**1**	50.70	63	*4*	0.84	0.79	0.81	16.60
ISB_DSC	5	*4*	0.94	0.82	0.88	4.70	21	*7*	0.68	0.74	0.55	0.90
ADPC	35	*2*	0.66	0.4	0.4	0.00	400	*7*	0.69	0.7	0.59	0.00
				Compound						cluto-t5.8k		
RNN_LDH	7	6	**1**	**0.99**	**1**	6.80	33	6	**0.87**	**0.83**	**0.85**	5.00
RNN_DSC	8	6	0.89	0.86	0.87	**0.00**	32	*7*	0.83	0.8	0.8	0.90
IS_DSC	10	*5*	1	1	1	31.60	62	*7*	0.99	0.98	0.99	20.50
ISB_DSC	8	*8*	0.97	0.92	0.97	1.80	39	*12*	0.85	0.85	0.84	2.10
ADPC	9	*7*	0.8	0.8	0.62	0.00	560	6	0.82	0.79	0.78	**0.00**
				Flame						cluto-t7.10k		
RNN_LDH	7	2	**1**	**1**	**1**	0.80	40	9	**0.92**	**0.92**	**0.93**	2.60
RNN_DSC	8	2	**1**	0.96	0.98	**0.00**	28	*10*	0.88	0.88	0.9	0.30
IS_DSC	4	2	0.7	0	−0.01	21.70	30	*6*	0.87	0.91	0.82	16.60
ISB_DSC	4	2	0.69	0.01	−0.01	1.30	18	*18*	0.86	0.87	0.83	1.50
ADPC	27	2	**1**	**1**	**1**	**0.00**	395	*10*	0.49	0.56	0.33	0.00
				Dim1024						cluto-t8.8k		
RNN_LDH	63	16	**1**	**1**	**1**	**0.00**	27	8	**0.96**	**0.93**	**0.96**	1.30
RNN_DSC	59	16	**1**	**1**	**1**	1.40	22	8	0.94	0.91	0.95	0.10
IS_DSC	63	16	**1**	**1**	**1**	39.60	30	*4*	0.68	0.78	0.56	12.00
ISB_DSC	58	16	**1**	**1**	**1**	6.30	10	*14*	0.98	0.96	0.98	2.10
ADPC	2	16	**1**	**1**	**1**	**0.00**	240	*9*	0.59	0.6	0.45	0.00
				Spiral						Jain		
RNN_LDH	2	3	**1**	**1**	**1**	**0.00**	16	2	**1**	**1**	**1**	0.50
RNN_DSC	2	3	**1**	**1**	**1**	**0.00**	15	2	**1**	**1**	**1**	**0.00**
IS_DSC	5	3	**1**	**1**	**1**	50.30	16	2	**1**	**1**	**1**	36.20
ISB_DSC	2	3	**1**	**1**	**1**	**0.00**	16	2	**1**	**1**	**1**	**0.00**
ADPC	13	3	**1**	**1**	**1**	**0.00**	38	2	0.59	0.18	−0.02	**0.00**

If the number of clusters (C) found out is not correct, it is in italics. The best benchmark is written in bold in the condition of right cluster number.

A particularly challenging feature of Frame, t7.10k, and t8.8k is that classes have homogeneous distributions and are very close to each other. RNN-LDH outperforms the other algorithms on the data sets. On the data set Frame, RNN-LDH takes two outliers in the upper-left corner as noise while ADPC classifies these two objects to the upper class. RNN-DSC misclassifies one object in the adjacent area of two classes. On t8.8k, the result of RNN-DSC is closed to RNN-LDH. Although ISB-DSC has the highest benchmarks, we can see it partition the data set incorrectly from Figure 3.

Spiral has 3 classes which embrace each other, and Dim1024 is a high-dimensional data set and has 16 Gaussian classes with 1024 points. From Table 2, we can see the clustering algorithms all can get good results, but IS-DSC has a high noise ratio. Jain has two moon shape classes with different densities. ADPC divides the high density class into two parts and classifies the lower density class to the nearest part. Results of these three data sets are not displayed.

t4.8k has six classes with random noise. A thin sine curve runs across classes. RNN-LDH partitions the data set into 8 clusters for the sine curve is divided into several segments: the upper-left segment and the bottom-right segment are treated as two clusters, the bottom-left segment is looked upon as noise, and other segments are classified into their nearest clusters. RNN-DSC detected out only one segment of this curve. The other three algorithms are unable to partition some of main classes.

t5.8k has six label-like classes and a thick stick running across them. It also contains random noise. All label-like classes are found by the five algorithms. IS-DSC gets the highest benchmarks with highest noise ratio again. Our algorithm treats the stick as noise. RNN-DSC finds out one segment of the stick. ISB-DSC finds out 3 segments of the stick as 3 independent clusters and classifies some noise into 3 independent clusters too. ADPC partitions all objects into 6 clusters.

4.2. Real-World Data. Table 3 shows the real-world data sets we used to test the algorithms, which were downloaded from website UCI [23]. For real-world data sets, it should be noted that we did a few data preprocessing on some of them or selected the subset from them to do experiments, which are all listed below:

(i) All samples with null or uncertain values or duplicates in the data sets were removed. Such data sets are Breast_C_W, Echocardiogram, and Internet-Ads.

(ii) Most of data sets have class attributes or character attributes. So Table 3 only shows the number of attributes used to compute distances of samples.

Table 3: Real-world data sets.

Data	Objects	Attributes	Classes
Breast_C_W	683	9	2
Internet-Ads	2359	1558	2
Image-seg	2100	19	7
Lung-cancer	32	56	3
SPECT-Heart	187	22	2
Zoo	101	16	7
Wine	178	13	3
Echocardiogram	106	11	2
Liver-disorders	345	6	2
Monks-3	432	6	2
Sonar	208	60	2
Ionosphere	351	34	2
Wholesale	440	8	3
Heart-disease	294	13	5
Contraceptive-M	1473	8	3
Hayes-roth	160	4	3

Table 4: Results of algorithm on real-world data sets.

Algorithm	k	C	F1	AMI	ARI	NR (%)	k	C	F1	AMI	ARI	NR (%)
				Breast_C_W						Internet-Ads		
RNN_LDH	30	2	**0.97**	**0.83**	**0.9**	11.00	130	2	**0.92**	**0.43**	**0.63**	13.80
RNN_DSC	11	2	0.69	0.01	−0.01	0.30	107	2	0.92	0.39	0.59	2.90
IS_DSC	—	—	—	—	—	—	—	—	—	—	—	—
ISB_DSC	78	2	0.97	0.81	0.87	8.70	146	2	0.91	0.38	0.58	2.50
ADPC	44	2	0.93	0.62	0.74	0.00	400	2	0.9	0.31	0.53	0.00
				Image-seg						Lung-cancer		
RNN_LDH	22	7	**0.65**	**0.59**	**0.41**	1.80	4	3	**0.63**	**0.15**	**0.15**	15.60
RNN_DSC	12	7	0.59	0.56	0.36	0.30	2	*1*	0.58	—	0	0.00
IS_DSC	33	*5*	0.71	0.64	0.49	39.90	7	*2*	0.62	0.12	0.13	50.00
ISB_DSC	19	*6*	0.65	0.58	0.41	2.80	7	*2*	0.69	0.28	0.29	0.00
ADPC	12	7	0.63	0.58	0.36	0.00	2	*1*	0.58	—	0	0.00
				SPECT-Heart						Zoo		
RNN_LDH	14	2	**0.89**	0	−0.02	0.40	9	7	**0.79**	0.73	0.66	28.70
RNN_DSC	5	*1*	0.96	0	0	0.40	4	*8*	0.77	0.7	0.6	7.90
IS_DSC	25	*1*	0.94	—	0	41.60	23	*3*	1	1	1	53.50
ISB_DSC	16	2	0.89	**0.09**	**0.27**	5.20	6	7	0.79	**0.77**	**0.72**	44.60
ADPC	10	2	0.82	0.02	0	0.00	12	7	0.63	0.58	0.36	0.00
				Lung-cancer						Liver-disorders		
RNN_LDH	4	3	**0.63**	**0.15**	**0.15**	15.60	9	2	**0.67**	0	**0**	0.00
RNN_DSC	2	*1*	0.58	—	0	0.00	5	2	0.67	0	0	0.30
IS_DSC	7	*2*	0.62	0.12	0.13	50.00	—	—	—	—	—	—
ISB_DSC	7	*2*	0.69	0.28	0.29	0.00	11	2	0.67	0	−0.01	5.20
ADPC	2	*1*	0.58	—	0	0.00	31	2	0.67	**0**	0	0.00
				Hayes-roth						Wine		
RNN_LDH	6	3	**0.59**	**0.18**	0.14	19.40	34	3	0.72	0.39	**0.38**	3.90
RNN_DSC	4	3	0.45	0.02	0.02	0.60	20	3	**0.73**	**0.42**	0.38	0.60
IS_DSC	13	*2*	0.67	0.05	0.05	34.40	16	3	0.7	0.31	0.29	38.20
ISB_DSC	8	3	0.58	0.16	**0.14**	28.80	12	*4*	0.64	0.41	0.34	3.40
ADPC	13	*2*	0.46	−0.01	−0.01	0.00	21	3	0.72	0.41	0.37	0.00
				Sonar						Monks-3		
RNN_LDH	11	2	**0.66**	0.01	**0**	4.30	5	2	**0.66**	0	0	1.90
RNN_DSC	6	2	0.57	0	0	2.40	3	*3*	0.65	0	0	0.50
IS_DSC	—	—	—	—	—	—	9	2	**0.66**	0	0	14.60
ISB_DSC	27	2	0.62	0	−0.01	5.30	7	2	**0.66**	−0.01	0	0.00
ADPC	14	2	0.66	**0.02**	0	0.00	10	2	0.65	**0.08**	**0.02**	0.00
				Echocardiogram						Heart-disease		
RNN_LDH	6	2	0.71	0.01	−0.03	2.30	6	4	**0.55**	0.02	0.03	4.00

Table 4: Continued.

Algorithm	k	C	F1	AMI	ARI	NR (%)	k	C	F1	AMI	ARI	NR (%)
RNN_DSC	4	2	0.71	0.01	0.06	0.80	4	4	0.53	0	−0.02	1.70
IS_DSC	28	2	**0.8**	**0.06**	**0.15**	47.00	—	—	—	—	—	—
ISB_DSC	23	2	0.7	0.06	0.15	2.30	7	*5*	0.53	0.03	0	7.60
ADPC	5	2	0.7	0.01	0.06	0.00	6	4	0.55	**0.02**	**0.04**	0.00
				Ionosphere						Contraceptive-M		
RNN_LDH	15	2	0.7	0.01	**0.02**	2.80	8	3	**0.51**	0.03	0	2.50
RNN_DSC	4	*3*	0.66	0.03	−0.05	2.60	4	*2*	0.52	0.01	0	0.20
IS_DSC	—	—	—	—	—	—	—	—	—	—	—	—
ISB_DSC	18	2	**0.83**	**0.02**	−0.08	37.90	7	*4*	0.52	0.01	0	6.40
ADPC	2	*4*	0.77	0.26	0.32	0.00	44	3	0.45	**0.03**	**0.02**	0.00

(iii) SPECT-Heart data set has two subsets, and we took the SPECT.test subset to test the algorithms.

(iv) All text values in Chess were replaced by numbers, such as "*f*" was replaced by 0 and "*t*" by 1 and so on.

(v) The attribute nos. 1 and 10–13 were removed from Echocardiogram, and the second attribute ("still-alive") was selected as the clustering label.

(vi) Lung-cancer is a sparse data set. There are 4 values for the fifth attribute, and 1 value for the ninth attribute was "?" (unknown). We replaced them with 0.

(vii) Heart-disease has 10 sub-data sets. We used "reprocessed hugarian data" to test the algorithms. This data set is also unbalance because its largest cluster has more than 60% samples, while the smallest one has less than 6% samples.

Table 4 shows the experiment results of the five methods.

The attribute characters of Internet-Ads, Echocardiogram, Heart-disease, and Liver-disorders are categorical, integral, and real. The first three data sets are unbalance data sets because their vast majority of samples are in one class. Internet-Ads are also sparse. The benchmarks show that RNN-LDH outperforms other algorithms on Internet-Ads. For Echocardiogram, IS_DSC gets the best benchmark, but it classifies near half samples into noise. Compared to the other 3 algorithms, RNN-LDH gets the best results on F1.

The attribute values of Breast_C_W, Lung-cancer, and Wholesale are all integral. Our algorithm outperforms the others on all benchmarks for the first two data sets. For Lung-cancer, the other four algorithms cannot get the correct cluster numbers.

The attribute characters of Image-seg, Wine, and Sonar are real. For Image-seg, RNN-LDH does the best work than the others. IS_DSC gets the highest benchmarks but with the highest noise ratio and the wrong cluster number. For Wine and Sonar, RNN-LDH outperforms the other algorithms on one benchmark.

The attribute characters of SPECT-Heart, Monk-3, and Hays-roth are categorical. SPECT-Heart is also unbalance. For these data sets, our method outperforms the other methods on F1. The attribute characters of the remaining data sets are multiple. Our method does better than RNN-DSC, IS-DSC, and ISB-DSC.

Table 5: Friedman test.

Data group	p value
RNN-LDH and RNN-DSC	$5.53e-06$
RNN-LDH and ISB-DSC	$9.90e-03$
RNN-LDH and APDC	$7.49e-07$

The experimental results of RNN-LDH are combined with the experimental results of RNN-DSC, ISB-DSC, and ADPC, respectively, into three data groups. Each data group has 2 columns and 135 rows. One column represents the algorithm RNN-LDH, and the other column is one of other three methods. 135 rows are divided into 5 labels: F1, AMI, ARI, NR, and CR. Label CR represents the correct ratio of cluster numbers, which is calculated by the following equation:

$$\mathrm{CR} = 1 - \left(\frac{|C - \mathrm{TC}|}{\mathrm{TC}}\right), \tag{6}$$

where C represents the cluster number the algorithms found out and TC represents the true cluster number of the data set.

The Friedman tests are carried out on these 3 data groups, and the p value of each test is listed in Table 5. Because the results of IS-DSC are not good, especially on the real-world data sets, we do not do the Friedman test on them with RNN-LDH.

The p values in Table 5 show that the results of our algorithm are significantly different with the results of the other algorithms.

5. Conclusions

In this paper, we proposed an improved density-based clustering algorithm, which is termed as RNN-DPC, by combining the k-reverse nearest neighbor model and the density hierarchical relationship. With the k-reverse nearest neighbor model, the proposed method partitions all observations of a data set into several unconnected core regions while outliers are around them initially. Comparing with density peak clustering, our method is more robust in finding initial clusters. By using the density hierarchical relationship, each unclustered object is grouped into the cluster that its parent object belongs to. If one's parent is itself or it is unclassified to any cluster, it is a noise. In

comparison with the RNN based method, our algorithm has lower noise ratio than IS-DSC and has higher accuracy than IS-DSC and RNN-DSC.

Data Availability

The data sets used in this paper are standard test data sets which are all available online and could be freely accessed. The synthetic data sets were downloaded from https://cs.joensuu.fi/sipu/datasets/and http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download, but the real-world data sets were downloaded from http://archive.ics.uci.edu/ml.

Conflicts of Interest

There are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by NSFC under Grant 61773022, Hunan Provincial Education Department (nos. 16B244, 17A200, and 18B504), and Natural Science Foundation of Hunan Province (nos. 2017JJ3287 and 2018JJ3479).

References

[1] N. Sunil Chowdary, D. Sri Lakshmi Prasanna, and P. Sudhakar, "Evaluating and analyzing clusters in data mining using different algorithms," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 2, pp. 86–99, 2014.

[2] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[3] T. Boongoen and N. Iam-On, "Cluster ensembles: a survey of approaches with recent extensions and applications," *Computer Science Review*, vol. 28, pp. 1–25, 2018.

[4] 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 96*, pp. 226–231, AAAI Press, Portland, OR, USA, August 1996.

[5] Y. H. Liu, Z. M. Ma, and F. Yu, "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," *Knowledge-Based Systems*, vol. 133, pp. 208–220, 2017.

[6] Y. Lv, T. Ma, M. Tang et al., "An efficient and scalable density-based clustering algorithm for datasets with complex structures," *Neurocomputing*, vol. 171, pp. 9–22, 2016.

[7] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[8] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted *K*-nearest neighbors," *Information Sciences*, vol. 354, pp. 19–40, 2016.

[9] J. Xu, G. Wang, and W. Deng, "DenPEHC: density peak based efficient hierarchical clustering," *Information Sciences*, vol. 373, pp. 200–218, 2016.

[10] Y.-A. Geng, Q. Li, R. Zheng, F. Zhuang, R. He, and N. Xiong, "RECOME: a new density-based clustering algorithm using relative KNN kernel density," *Information Sciences*, vol. 436-437, pp. 13–30, 2018.

[11] R. H. Liu, W. P. Huang, Z. S. Fei, K. Wang, and J. Liang, "Constraint-based clustering by fast search and find of density peaks," *Neurocomputing*, vol. 330, pp. 223–237, 2019.

[12] X. Xu, S. Ding, and Z. Shi, "An improved density peaks clustering algorithm with fast finding cluster centers," *Knowledge-Based Systems*, vol. 158, pp. 65–74, 2018.

[13] Y. Chen, S. Tang, L. Zhou et al., "Decentralized clustering by finding loose and distributed density cores," *Information Sciences*, vol. 433-434, pp. 510–526, 2018.

[14] J. Xie, Z.-Y. Xiong, Y.-F. Zhang, Y. Feng, and J. Ma, "Density core-based clustering algorithm with dynamic scanning radius," *Knowledge-Based Systems*, vol. 142, pp. 58–70, 2018.

[15] S. Vadapalli, S. R. Valluri, and K. Karlapalem, "A simple yet effective data clustering algorithm," in *Proceedings of the 6th International Conference on Data Mining*, pp. 1108–1112, Hong Kong, December 2006.

[16] A. Bryant and K. Cios, "RNN-DBSCAN: a density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1109–1121, 2018.

[17] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, and A. Pulvirenti, "Enhancing density-based clustering: parameter reduction and outlier detection," *Information Systems*, vol. 38, no. 3, pp. 317–330, 2013.

[18] M. A. Rahman, K. L.-M. Ang, and K. P. Seng, "Unique neighborhood set parameter independent density-based clustering with outlier detection," *IEEE Access*, vol. 6, no. 1, pp. 44707–44717, 2018.

[19] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation," *Journal of Machine Learning Research*, vol. 2, no. 1, pp. 37–63, 2011.

[20] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.

[21] University of Eastern Finland, *Clustering Datasets: Shape Sets*, University of Eastern Finland, Eastern Finland, Finland, 2016, https://cs.joensuu.fi/sipu/datasets/.

[22] G. Karypis, E. H. Han, and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.

[23] M. Lichman, "UCI machine learning repository," 2016, http://archive.ics.uci.edu/m.