Computers, Materials & Continua
DOI:10.32604/cmc.2021.012850
Article

Tech Science Press

Information Theoretic Weighted Fuzzy Clustering Ensemble

Yixuan Wang[1], Liping Yuan[2,3], Harish Garg[4], Ali Bagherinia[5], Hamïd Parvïn[6,7,8,*], Kim-Hung Pho[9] and Zulkefli Mansor[10]

[1]Department of Electronic Information Engineering, School of Information Engineering, Wuhan University of Technology, Wuhan, China
[2]School of Information Engineering, Wuhan University of Technology, Wuhan, China
[3]School of Information Engineering, Wuhan Huaxia University of Technology, Wuhan, China
[4]School of Mathematics, Thapar Institute of Engineering and Technology, Deemed University, Patiala, Punjab, India
[5]Department of Computer Science, Dehdasht Branch, Islamic Azad University, Dehdasht, Iran
[6]Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam
[7]Faculty of Information Technology, Duy Tan University, Da Nang, 550000, Vietnam
[8]Department of Computer Science, Nourabad Mamasani Branch, Islamic Azad University, Mamasani, Iran
[9]Fractional Calculus, Optimization and Algebra Research Group, Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[10]Fakulti Teknologi dan Sains Maklumat Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
[*]Corresponding Author: Hamid Parvin. Email: parvin@iust.ac.ir

Received: 14 July 2020; Accepted: 19 August 2020

Abstract: In order to improve performance and robustness of clustering, it is proposed to generate and aggregate a number of primary clusters via clustering ensemble technique. Fuzzy clustering ensemble approaches attempt to improve the performance of fuzzy clustering tasks. However, in these approaches, cluster (or clustering) reliability has not paid much attention to. Ignoring cluster (or clustering) reliability makes these approaches weak in dealing with low-quality base clustering methods. In this paper, we have utilized cluster unreliability estimation and local weighting strategy to propose a new fuzzy clustering ensemble method which has introduced Reliability Based weighted co-association matrix Fuzzy C-Means (RBFCM), Reliability Based Graph Partitioning (RBGP) and Reliability Based Hyper Clustering (RBHC) as three new fuzzy clustering consensus functions. Our fuzzy clustering ensemble approach works based on fuzzy cluster unreliability estimation. Cluster unreliability is estimated according to an entropic criterion using the cluster labels in the entire ensemble. To do so, the new metric is defined to estimate the fuzzy cluster unreliability; then, the reliability value of any cluster is determined using a Reliability Driven Cluster Indicator (*RDCI*). The time complexities of RBHC and RBGP are linearly proportional with the number of data objects. Performance and robustness of the proposed method are experimentally evaluated for some benchmark datasets. The experimental results demonstrate efficiency and suitability of the proposed method.

Keywords: Fuzzy clustering ensemble; cluster unreliability; consensus function

1 Introduction

Data mining [1–24] involves many tasks. One of the most important tasks in data mining is clustering [25–29]. According to different similarity criteria implemented by various clustering algorithms in the context of unsupervised learning, different objective functions are targeted [30]. According to "no free lunch" theory, there is no dominant clustering method [22,23]. Therefore, the idea of combining clustering, which is also called cluster (or clustering) ensemble, emerged. In a clustering ensemble, some of the basic partitions are combined to acquire a better solution capable of managing all objectives of different partitions which sometimes are contradictory [28–30]. Cluster ensemble offers many advantages, among which the following can be mentioned: robustness to noise [29], capability of producing novel results [31–33], quality enhancement [29], knowledge reusability [30], multi-view clustering [33], stability, parallel/distributed data processing [30], finding number of real clusters, adaptability and heterogeneous data clustering.

The process of clustering ensemble consists of two related phases as shown in Fig. 1a. The first phase includes production of the diverse base clusterings through the basic clustering algorithm (s), and the second phase involves extracting the final clustering from the primary basic partitions using a consensus function. To improve the performance of the clustering ensemble in the first phase, a special attention should be paid to the ensemble diversity which determines the quality of the combination. If the quality of each voter is greater than a random voter, the quality of the combination is improved by increasing the ensemble diversity, where the combination can be any community like the ensemble clustering [33]. There are many approaches, which are used in production of primary basic partitions in order to acquire the desired diversity, such as: Different initializations of parameters [30], heterogeneous ensemble clustering methods [28], different subsets of the features [30], different subsets of objects [28], projection of the object subsets [28] and hybrid methods. In our proposed method, a basic clustering algorithm is implemented with different parameters to achieve the desired diversity. In the second phase, a consensus function is utilized to achieve the final clustering. Selection of the clusters with high quality is an NP hard problem because clustering is an unsupervised problem [33]. Several consensus functions are proposed to face this issue, each of which uses a specific approach and different information from the primary basic partitions achieved from phase one, and sometimes considers the initial characteristics of the data. Consensus ensemble methods are divided into the following classes: 1. intermediate space clustering ensemble methods [28], 2. methods based on the co-association matrix [33], 3. hyper-graph based methods [30], 4. expectation maximization clustering ensemble methods [32], 5. mathematical modeling (median partition) approaches [34], and 6. voting-based methods [35].

There are two types of clustering algorithms: (a) First, hard clustering algorithms in which a data object either is definitely assigned to a cluster or is not assigned to it at all and (b) Second, fuzzy clustering algorithms in which there are some data points that are not allocated to a specific cluster but they are allocated to all clusters with different membership degrees (for a data point, the summation of all membership degrees to all clusters should be one). The basis of fuzzy clustering algorithms is the basic fuzzy c-means (FCM) clustering algorithm [36].

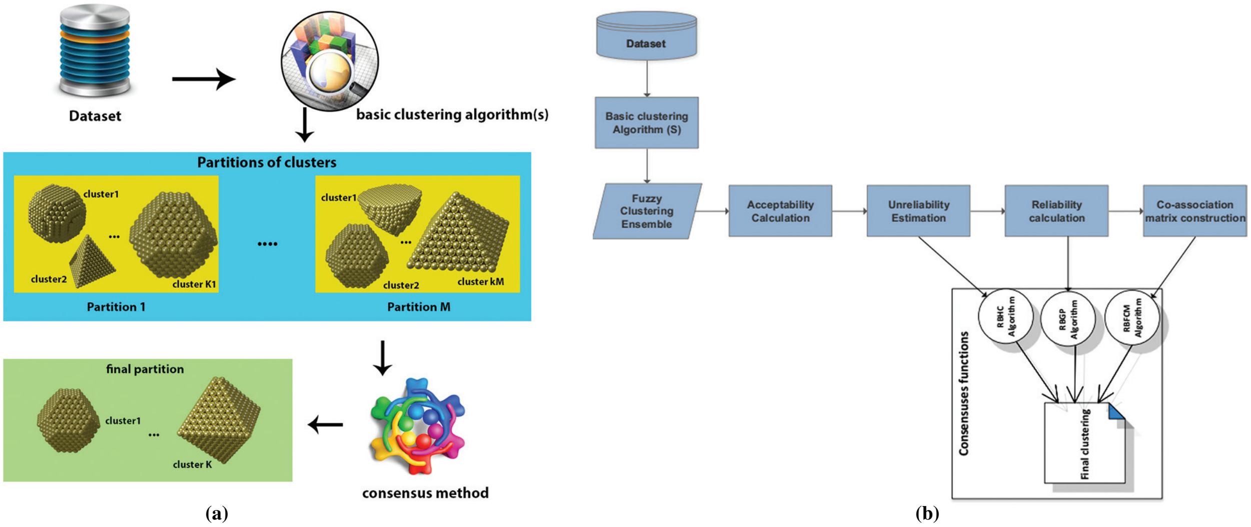

Figure 1: (a) The general procedure of a clustering ensemble method, and (b) The proposed approach steps

Although soft (or fuzzy) clustering has more generality than crisp clustering, researches in soft clustering are in early stages and fuzzy clustering ensemble approaches have not been widely developed. Fuzzy clustering ensembles are proposed by a few researchers, (e.g., Punera et al. [37] have proposed soft versions of Cluster-based Similarity Partitioning Algorithm (CSPA), Meta CLustering Algorithm (MCLA) and Hybrid Bipartite Graph Formulation (HBGF) which are respectively named soft CSPA (sCSPA), soft MCLA (sMCLA) and soft HBGF (sHBGF)) but crisp clustering has more maturity. Some of the existing fuzzy clustering ensemble methods convert fuzzy clusters to hard clusters through some existing simple methods and then the hard consensus functions are used to compute the final partition. This conversion causes the loss of uncertainty information. This loss of uncertainty information causes some challenges in suggesting an efficient fuzzy consensus clustering from multiple fuzzy basic partitions.

Consensus process is highly dependent on the quality of the primary partitions so that the low-quality or even noisy primary partitions could adversely affect it. Improving performance of the consensus functions through quality-evaluation and weighting of the primary partitions is attended to confront with low-quality primary partitions based on the implicit assumption that all clusters have the same reliability value in the same base clustering [38]. This is usually performed by assigning a weight to each primary partition, which is considered to be an individual as a whole, without caring about the quality of its clusters. We have to confront different reliability values of clusters in the same clustering because real-world datasets are inherently complex and noisy. In some methods, it is needed to have access to the data features such as what was proposed by Zhong et al. [33]. Their method has investigated the reliability values of the clusters using the Euclidean distances between objects of the clusters and its efficiency is highly dependent on the distribution of the data in the dataset. However, in a clustering ensemble in its general formulation, we assume that we have not access to the original data features.

In this work, according to estimation of ensemble driven cluster unreliability and local weighting strategy, a new approach of fuzzy clustering ensemble is proposed as shown in Fig. 1b. In order to increase the consensus performance, a locally weighted plan is achieved by integrating the validity and unreliability of the cluster. An entropic criterion is used to estimate unreliability of any fuzzy cluster based on relation of the cluster acceptability to the entire ensemble. The new metric is defined to estimate the fuzzy cluster unreliability and then the reliability values of the clusters are determined using an index named *RDCI*. Each cluster is assessed and weighted by its *RDCI* which evaluates the cluster using an effective indication provided by the crowd of diverse clusters in the ensemble. Then, a fuzzy weighted co-association matrix with the calculated weights based on the reliability, incorporates local acceptability into the conventional co-association (Co) matrix and is treated as a summary for the ensemble of diverse clusters. After that, new consensus algorithms are proposed to achieve the final clustering. The new consensus algorithms include: (a) RBFCM and (b) RBGP which consider cluster reliabilities, and (c) RBHC which considers cluster pairwise acceptability.

The main contributions of the paper are summarized as follows. (I) We estimate an unreliability value per a fuzzy cluster in relation to other clusterings through the proposed method utilizing an entropic criterion. The entropic criterion considers only distribution of membership degrees of all objects to each primary fuzzy cluster while the access to the original data features is not needed; it has also no assumptions on data distribution. (II) The primary fuzzy clusters in the ensemble are assessed and weighted by proposing an *RDCI* which results in providing a reliability indication at the cluster-level with a contribution to the local weighting plan. (III) A new approach for fuzzy co-association matrix computation in the fuzzy cluster ensemble is proposed. (IV) New consensus functions are proposed to construct the final clustering according to estimation of the reliability–driven fuzzy cluster unreliability and local weighting strategy. (V) Finally, the derived experimental results demonstrate the performance and robustness superiority of the proposed fuzzy clustering ensemble approach to the state-of-the-art approaches.

The rest of the paper is organized as follows. In Section 2, the related literature is reviewed. Section 3 provides the background knowledge about entropy and clustering ensemble. The proposed approach of fuzzy clustering ensemble is described in Section 4 based on the cluster unreliability estimation and local weighting strategy. The experimental results are provided in Section 5, and finally the paper is concluded in Section 6.

2 Related Work

Some of the most important ensemble clusterings include k means-based consensus clustering [28], spectral ensemble clustering [39] and infinite ensemble clustering [40]. The following works are also considered to be important researches in fuzzy clustering ensemble: sCSPA and sMCLA introduced in [37] as the fuzzy extension versions of CSPA and MCLA are proposed in [30]. sHBGF is also proposed as a fuzzy version of HBGF [31].

To extract the final fuzzy clustering out of a fuzzy clustering ensemble, an explicit objective function is proposed in [41] based on the new contingency matrix. In order to accurately and efficiently extract the final clustering in this approach which is a parallelizable algorithm and capable of being used for big data clustering, a flexible utility function is employed to change fuzzy consensus clustering into a weighted piecewise FCM-like iterative.

Vote-based merging algorithm (VMA) as a fuzzy clustering ensemble method which was proposed in [42], calculated the final clustering based on the membership matrix averaging in each clustering. Since all the base clusters in this method have to be relabeled, VMAs would be among time-consuming algorithms. An information theoretic kmeans (ITK) was introduced by Dhillon et al. [43].

In [34], the clusters labels are shown as a 0–1 bit string where the goal is to obtain the final fuzzy clustering which is a membership matrix any of whose members shows the membership degree of a data object to a cluster. Accordingly, they introduce the Fuzzy String Objective Function (FSOF) which tries to minimize the summation of distances between centers of the initial clusters and the final clusters. Since the final clustering was fuzzy, some constraints were added to the objective function which created a non-linear optimization NP problem. They employed a genetic algorithm for its solution. Note that it is applicable while the base clusterings are crisp. Two consensus functions, i.e., Fuzzy String Cluster Ensemble Optimized by Genetic Algorithm numbers 1 and 2 (FSCEOGA1 and FSCEOGA2), were also proposed based on cross-over and mutation operators.

In order to enhance the stability of the fuzzy cluster analysis, a heterogeneous clustering ensemble has been proposed [44] in which basic fuzzy clustering algorithms are first utilized and then, final clustering is obtained using a FCM algorithm so that all clusters in the co-association matrix have equal participation weights. To achieve the consensus clustering from fuzzy clustering ensembles, a voting mechanism is used in [35]. The work includes disambiguation and voting procedures. Disambiguation is a phase in which Hungarian algorithm [45] with time complexity of $O\left(\overline{n}^3\right)$ ($\overline{n}$ is (average) number of clusters in each clustering) is used for re-labeling problem. The voting phase is implemented to achieve the final consensus clustering. There are many voting phases such as confidence-based voting methods (including sum voting rule and product voting rule) and positional-based voting methods (including Borda voting rule and Copeland voting rule). Time complexity of these voting rules is $O(\overline{n}M\beta)$ where $\overline{n}$, M and β are (average) number of the clusters in each clustering, number of data objects and number of base clusterings respectively. There exist many different consensus functions depending on direct or repetitive combination of re-labeling and voting phases.

According to the particle swarm optimization (PSO) which has the capability of finding fuzzy and crisp clusters [46,47], a method for construction of a fuzzy clustering ensemble is proposed in [46]. It creates initial clusters through parameter change; and then, using pruning process, n' clusters are chosen from the β initial clusterings so that $n' \ll c$ where c is the number of the clusters in all of the β initial clusterings. For pruning process, one of the internal cluster validity indices such as Ball–Hall, Caliński et al. [48], Dunn index [49], Silhouette index [50] or Xie-Beni [51] is selected to evaluate the fitness of primary basic clusters; and then, one of the genetic selection mechanisms such as tournament or roulette wheel is used to choose the elite clusters. Final clustering is subsequently achieved by use of the consensus function through PSO algorithm. Unlike other PSO-based methods in which each particle represents a clustering, each particle represents a cluster in this method.

3 Preliminaries

Through the following definitions, we introduce the general formulation of data, fuzzy clustering ensemble and entropy in this paper. Note that the notations implemented in this work are presented here. An object (or data point) is a tuple denoted by $x_i = \left(x_i^1, x_i^2, \ldots x_i^N\right)$. It is also a vector where x_i^j represents the j-th attribute from i-th data. Also, $x_:^j$ is defined as j-th attribute of dataset, $N = \left| x_1^: \right|$ denotes size of the dataset dimensions (i.e., quantity of the attributes), and $M = \left| x_:^i \right|$ shows the dataset size (i.e., quantity of the data points). $\pi(x)$ is a $M \times n$ matrix which represents the fuzzy partition defined on dataset x, where $M = \left| x_:^1 \right|$ and n is an integer indicating the number of clusters, and we have Eq. (1).

$$\forall j \in \{1, \ldots, n\}, \quad i \in \{1, \ldots, M\} : \pi\,(x_i)^j \in [0, 1] \tag{1}$$

$$\forall i \in \{1, \ldots, M\} : \sum_{j=1}^{n} \pi\,(x_i)^j = 1$$

where $\pi\,(x_i)^j$ shows a real number indicating how much i-th data object belongs to the j-th cluster in partition π which is $\pi = \{C_1, C_2, \ldots, C_n\}$. Π denotes a clustering ensemble which includes β primary partitions, i.e., $\Pi = \left\{\pi^1, \ldots, \pi^\beta\right\}$ in which $\pi^m = \left\{C_1^m, \ldots, C_{n^m}^m\right\}$ where, π^m represents the m-th primary partition in Π, C_i^m and n^m are the i-th cluster and the number of the clusters in π^m respectively. The set of all clusters in the ensemble is denoted by $\boldsymbol{C}$ and defined as $\boldsymbol{C} = \left\{C_1^1, \ldots, C_{n^1}^1, \ldots, C_1^\beta \ldots, C_{n^\beta}^\beta\right\}$ where C_i^j is the i-th cluster of the partition π^j. Therefore, the number of all clusters in the base clusterings is denoted by c and is defined as $c = n^1 + \ldots + n^\beta$. Let's assume that z is a discrete random variable. The entropy is a measure of the unreliability related to a random variable. For a discrete random variable z whose domain, i.e., the set of the values for z, is denoted by Z, it is defined according to Eq. (2).

$$H\,(z) = -\sum_{z' \epsilon Z} \rho\left(z = z'\right) \log_2 \rho\left(z = z'\right) \tag{2}$$

where $\rho\left(z = z'\right)$ is the probability mass function of z and $0\log_2 0$ is always assumed to be 0 here. $H\,(z, w)$ is the joint entropy as a measure of the unreliability associated with a set of two discrete random variables z and w. It is defined according to Eq. (3).

$$H\,(z, w) = -\sum_{z' \in Z} \sum_{w' \in W} \rho\left(z = z', w = w'\right) \log_2 \rho\left(z = z', w = w'\right) \tag{3}$$

where $\rho\,(z, w)$ is the joint probability of two discrete random variables z and w. When z and w are independent random variables, the joint entropy is $H\,(z, w) = H\,(z) + H\,(w)$. Thus, for q independent random variables $z_1, z_2, \ldots, z_q$, the joint entropy is defined according to $H\left(z_1; \ldots; z_q\right) = \sum_{q'=1}^{q} H\left(z_{q'}\right)$.

4 Proposed Approach

Block diagram of the proposed method is presented in Fig. 1b in which a new approach of fuzzy clustering ensemble is proposed based on estimation of the ensemble driven cluster unreliability and local weighting strategy. The proposed algorithm includes the following stages. First, we compute the acceptability of each cluster. Then, the unreliability of each cluster is estimated and after that weight of each cluster is determined. Next, the weighted co-association matrix is produced and finally, the consensus clustering is achieved.

4.1 Cluster Acceptability Computation

According to Fig. 1b and Eq. (4), the acceptability of each cluster over other clusters is computed which is equivalent to the computation of agreement probability between two clusters in different clusterings. Acceptability computation of cluster C_i^s (cluster $C_i \in \pi^s$) over the cluster C_j^r (cluster $C_j \in \pi^r$), when $s \neq r$ is performed using Eq. (4).

$$\rho\left(C_i^s, C_j^r\right) = 1 - \frac{\sum_{t=1}^{M}\left|\left(\pi^s\left(x_t\right)^i - \pi^r\left(x_t\right)^j\right)\right|}{M} \tag{4}$$

4.2 Cluster Unreliability Estimation

As depicted in Fig. 1b, the next stage in the algorithm is unreliability estimation of the clusters. Using the entropy concept, as a measure for discrete random variables, unreliability is computed to be applied to the cluster labels in reliability computation of the clusters. It is a reasonable strategy because there is not information about original data features and their distribution. If cluster C_i does not belong to the base clustering $\pi^r \in \Pi$ exactly, then, the clustering π^r will partition the cluster; this means that the membership degree of data objects of C_i, is different from their membership degree in all clusters of π^r and the clustering π^r may not accept the membership degree of C_i data objects. Therefore, depending on how the data objects of C_i are clustered in π^r, the unreliability (or entropy) of C_i with regards to π^r is estimated by entropy concept presented by Eq. (2). Unreliability of the cluster C_i^s (i.e., the i-th cluster in s-th clustering) with respect to clustering π^r of the ensemble Π where $s \neq r$, is acquired using Eq. (5).

$$\upsilon\left(C_i^s, \pi^r\right) = \frac{-\sum_{j=1}^{n^r}\dot{\rho}\left(C_i^s, C_j^r\right)\log_2\dot{\rho}\left(C_i^s, C_j^r\right)}{\log_2 n^r} \tag{5}$$

where $\dot{\rho}\left(C_i^s, C_j^r\right)$ is the probability vector obtained according to $\dot{\rho}\left(C_i^s, C_j^r\right) = \frac{\rho\left(C_i^s, C_j^r\right)}{\sum_{t=1}^{n^r}\rho\left(C_i^s, C_t^r\right)}$ and the term $\log_2 n^r$ is added to guarantee that $\upsilon\left(C_i^s, \pi^r\right)$ is in range $[0; 1]$, where n^r is the number of the clusters in π^r, C_j^r is the j-th cluster in π^r and $\rho\left(C_i^s, C_j^r\right)$ is acquired according to Eq. (4). Since we have assumed that the partitions of the ensemble are independent [52], we can achieve the unreliability of a cluster C_i with respect to the β base clusterings in ensemble Π by Eq. (6). It means the unreliability of cluster C_i^s in ensemble clusterings Π is computed by Eq. (6).

$$\mho\left(C_i^s, \Pi\right) = \frac{\sum_{j \in \{1,2,\ldots,\beta\}-\{s\}}\upsilon\left(C_i^s, \pi^j\right)}{\beta - 1} \tag{6}$$

where the term $\beta-1$ is added to make $\mho\left(C_i^s, \Pi\right)$ in the range $[0; 1]$.

4.3 Cluster Reliability Computation

Weight of each cluster represents its reliability value which is computed according to the derived unreliability or entropy of that cluster in the clustering ensemble via an *RDCI*. For cluster C_i^s, *RDCI* as the weight of each cluster in a clustering ensemble Π with β base clusterings is defined as follows in Eq. (7).

$$RDCI_{\varnothing}\left(C_i^s\right) = \exp\left(\frac{-\mho\left(C_i^s, \Pi\right)}{\varnothing}\right) \tag{7}$$

where impact of the unreliability on clustering ensemble weight is adjusted by the non-negative parameter $\varnothing$. The best result is obtained when $\varnothing$ is 0.4. Since $\mho\left(C_i^s, \Pi\right) \in [0, 1]$, $RDCI_{\varnothing}\left(C_i^s\right)$ lies in the interval $\left[e^{\frac{-1}{\varnothing}}, 1\right]$ for each $C_i^s \in \Pi$. When the unreliability value of a cluster C_i^s is minimized, i.e., when $\mho\left(C_i^s, \Pi\right)$ is zero, its $RDCI$ is maximized, i.e., $RDCI_{\varnothing}\left(C_i^s\right)$ is one.

4.4 Cluster-Wise Weighted Co-Association Matrix

In this stage, a fuzzy co-association matrix is derived regarding the reliability values of the clusters in the ensemble. Methods based on the co-association matrix are of the most common methods for combination of the base clusterings. Using the Evidence Accumulation Clustering (EAC) method, which was proposed in [32], individual data object clusterings in a clustering ensemble were projected into a new metric of pairwise similarity. However, this method could not suitably achieve the co-association matrix from fuzzy clusters. Thus, Evidence Accumulation Fuzzy Clustering (EAFC) is implemented as a new method to derive the co-association matrix. Eq. (8) is used to derive the fuzzy co-association clustering ensemble matrix

$$\dot{FCA}_{i,j}^{\Pi} = \frac{\sum_{k=1}^{\beta} \overset{n^k}{\underset{t=1}{}} \sup \inf \left(\pi^k \left(x_i\right)^t, \pi^k \left(x_j\right)^t\right)}{\beta} \tag{8}$$

where x_i and x_j are the data objects, $\inf(x, y)$ and $\sup(x, y)$ are considered to be xy and $x + y$ respectively. To make the $\dot{FCA}_{i,j}^{\Pi}$ matrix normal, we use $FCA_{i,j}^{\Pi} = \frac{\dot{FCA}_{i,j}^{\Pi}}{\sqrt{\dot{FCA}_{i,i}^{\Pi} \times \dot{FCA}_{j,j}^{\Pi}}}$. In Eq. (9), the weighted fuzzy co-association matrix is obtained by using the $RDCI$ as a weight in calculation of the co-association matrix to consider the reliability of each cluster. Calculation of the weighted fuzzy co-association clustering ensemble matrix ($WFCA$) is performed using Eq. (9).

$$W\dot{FCA}_{i,j}^{\Pi} = \frac{\sum_{k=1}^{\beta} \overset{n^k}{\underset{t=1}{}} \sup RDCI_{\varnothing}\left(C_t^k\right) \times \inf \left(\pi^k \left(x_i\right)^t, \pi^k \left(x_j\right)^t\right)}{\beta} \tag{9}$$

To make the $W\dot{FCA}_{i,j}^{\Pi}$ matrix normal, we use $WFCA_{i,j}^{\Pi} = \frac{W\dot{FCA}_{i,j}^{\Pi}}{\sqrt{W\dot{FCA}_{i,i}^{\Pi} \times W\dot{FCA}_{j,j}^{\Pi}}}$.

Toy example. Tab. 1a represents two fuzzy clusterings π^1 and π^2 (i.e., $\beta = 2$) on an assumptive dataset x with 6 data objects (i.e., $M = 6$). Tab. 1b represents the ρ values of the fuzzy clusters in Tab. 1a. Tab. 1c shows $\dot{\rho}$ values of the fuzzy clusters in Tab. 1a and their unreliability values. Tab. 1c also contains the corresponding $RDCI$ values of fuzzy clusters. Using Eq. (8), the co-association matrix FCA of π^1 and π^2 from Tab. 1a is derived and shown in Tab. 1d. Tab. 1e exhibits the weighted co-association matrix of the fuzzy clustering ensemble presented in Tab. 1a regarding the calculated $RDCI$s.

Table 1: An example for the proposed method

(a)

	π_1			π_2		
	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2	C_3^2
x_1	0.2	0.8	0	0.3	0.6	0.1
x_2	0.1	0.1	0.8	0.2	0.1	0.7
x_3	0.3	0.2	0.5	0.3	0.3	0.4
x_4	0.1	0.9	0	0.2	0.7	0.1
x_5	0.7	0.2	0.1	0.7	0.1	0.2
x_6	1	0	0	0.9	0.1	0

(b)

	π_1			π_2		
	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2	C_3^2
C_1^1	–	–	–	0.93	0.58	0.62
C_2^1	–	–	–	0.53	0.88	0.62
C_3^1	–	–	–	0.53	0.62	0.92
C_1^2	0.93	0.53	0.53	–	–	–
C_2^2	0.58	0.88	0.62	–	–	–
C_3^2	0.62	0.62	0.92	–	–	–

(c)

	π_1			π_2			$\mho\left(C_i^s, \Pi\right)$	$RDCI_{0.4}\left(C_i^s\right)$
	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2	C_3^2		
C_1^1	–	–	–	0.44	0.27	0.29	0.979	0.0865
C_2^1	–	–	–	0.26	0.43	0.31	0.979	0.0865
C_3^1	–	–	–	0.26	0.30	0.44	0.974	0.0876
C_1^2	0.47	0.27	0.27	–	–	–	0.965	0.0896
C_2^2	0.28	0.42	0.30	–	–	–	0.984	0.0854
C_3^2	0.29	0.29	0.43	–	–	–	0.983	0.0856

(d)

	π_1			π_2		
	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2	C_3^2
x_1	1.00	0.25	0.59	0.99	0.53	0.37
x_2	0.25	1.00	0.88	0.22	0.40	0.20
x_3	0.59	0.88	1.00	0.53	0.70	0.52
x_4	0.99	0.22	0.53	1.00	0.40	0.22
x_5	0.53	0.40	0.70	0.40	1.00	0.96
x_6	0.37	0.20	0.52	0.22	0.96	1.00

(Continued)

Table 1: (continued)

(e)

	π_1			π_2		
	C_1^1	C_2^1	C_3^1	C_1^2	C_2^2	C_3^2
x_1	1.00	0.25	0.58	0.99	0.54	0.38
x_2	0.25	1.00	0.88	0.22	0.40	0.20
x_3	0.58	0.88	1.00	0.52	0.70	0.53
x_4	0.99	0.22	0.52	1.00	0.40	0.23
x_5	0.54	0.40	0.70	0.40	1.00	0.96
x_6	0.38	0.20	0.53	0.23	0.96	1.00

4.5 Consensus Functions

Final clustering computation, which is the last stage according to Fig. 1b, is performed through three proposed consensus algorithms in this section based on the following ways: (a) Using co-association matrix and a subsequent hierarchical clustering, (b) According to reliability computation of each cluster of the ensemble and a subsequent graph clustering, and (c) Based on the acceptability computation of each cluster over other clusters of the ensemble and subsequently applying of a metaheuristic algorithm.

4.5.1 WFCA-Based Consensus Function

When a cluster's unreliability is a large value with respect to the base clusterings, we understand that the cluster is divided into some fragments of the data objects in the partitions of the ensemble. Here, a conclusion is to keep the clusters with large reliability values with respect to the clustering ensemble in the final ensemble of elite clusters. By considering reliability values of clusters in co-association matrix computation, and viewing it as a new similarity matrix between the data object pairs in the dataset, the consensus partition can be achieved by applying a simple FCM clusterer algorithm or a hierarchical clusterer algorithm over the new similarity matrix.

We need to calculate the weight of each cluster using Eq. (7) to achieve the final clustering from base clusterings Π based on each cluster's local reliability value in the ensemble. Therefore, to compute the entropy of each cluster in Π, the following steps must be followed: the acceptability of any cluster (denoted by p) in the ensemble Π is obtained by Eq. (4) and then, it is normalized. Then, the unreliability of each cluster is computed by Eq. (5) with respect to the clustering π^m. Subsequently, the entropy of each cluster is calculated according to Eq. (6) in the ensemble. After computation of the *RDCI*, using Eq. (9), the weighted fuzzy co-association matrix of the clustering ensemble is acquired and then, it is normalized. For consensus function, a hierarchical clustering algorithm is implemented which is a widespread clustering technique whose typical input is a distance matrix d achieved by $d = 1 - WFCA$. Also, for consensus function, a simple FCM clustering algorithm can be employed instead of the hierarchical clustering algorithm by considering the distance matrix d as an intermediate feature space. RBFCM algorithm with two inputs (primary clustering ensemble Π and n^* number of clusters in the consensus partition) is presented in detail as follows in Algorithm 1.

Algorithm 1:: RBFCM (Reliability Based Weighted Co-association matrix Algorithm)

```
Input: $\mathbf{\Pi}$, $n^*$;
Output: $\boldsymbol{\pi}^*$
// $\mathbf{\Pi}$ stands for a pool of fuzzy primary partitions
// $n^*$ stands for number of final clusters
// $\boldsymbol{\pi}^*$ stands for final clustering
    According to Eq. (4), the cluster acceptability over other clusters is calculated in clustering
      $\pi^r$ ($\pi^r \in \mathbf{\Pi}$) for each cluster.
    According to Eq. (5), the cluster unreliability is computed with respect to the clustering
      $\pi^m$ in the ensemble $\mathbf{\Pi}$ for each cluster.
    Using Eq. (6), the unreliability of the clusters in the ensemble $\mathbf{\Pi}$ is derived.
    Using Eq. (7), $RDCI$ values of the clusters in ensemble are calculated.
    According to Eq. (9), $WFCA$ matrix is created.
    Using $d = 1 - WFCA$, $d$ matrix is computed.
    Final clustering with $n^*$ clusters is achieved using $FCM(d, n^*)$ //$\boldsymbol{\pi}^* = \mathrm{FCM}(d, n^*)$
Output: Consensus clustering $\boldsymbol{\pi}^*$.
```

4.5.2 *Reliability Based Graph Clustering Algorithm*

According to the definition of the bipartite graph and its clustering in which all of clusters and data points are considered to be its nodes, RBGC consensus function is proposed. In order to have an edge between two vertices, they have to be different. Thus, if one node is a data object and the other is a cluster, an edge could be developed. Weight of an edge located between a data object $x_i \in X$ and a cluster $v_j \in C$ is given by Eq. (10).

$$E(i,j) = \begin{cases} RDCI_{\varnothing}\left(v_j\right) \times v_j\left(x_i\right) & x_i \in X,\ v_j \in C \\ E(j,i) & x_j \in X,\ v_i \in C \\ 0 & otherwise \end{cases} \tag{10}$$

where $RDCI_{\varnothing}\left(v_j\right)$ is given by Eq. (7) and $v_j\left(x_i\right)$ is a real number in $[0;1]$ which indicates how much data object x_i belongs to in cluster v_j. The result of application of $RDCI$s is, in addition of considering the belong-to relationship between objects and clusters by bipartite graph, the reliability values of clusters are reflected. In order to obtain the consensus clustering, after the weighted bipartite graph was constructed, some basic graph partitioning algorithms such as METIS [53] are utilized to divide the mentioned weighted bipartite graph into n^* clusters, where n^* is the quantity of the desired clusters in the consensus partition and the consensus graph partition is considered to be consensus partition. Algorithm 2 shows RBGP algorithm in detail.

Algorithm 2:: RBGP (Reliability Based Graph Partitioning Algorithm)

```
Input: $\mathbf{\Pi}$, $n^*$;
Output: $\boldsymbol{\pi}^*$
// $\mathbf{\Pi}$ stands for a pool of fuzzy primary partitions
// $n^*$ stands for number of final clusters
// $\boldsymbol{\pi}^*$ stands for final clustering
```

(Continued)

Algorithm 2:: (Continued)

```
Using Eq. (4), the acceptability of each cluster is calculated over other clusters in the
  clustering $\pi^m$ of the ensemble $\mathbf{\Pi}$.
Eq. (5) is implemented to compute the unreliability of each cluster with respect to the
  partition $\pi^m$ in the ensemble $\mathbf{\Pi}$.
The unreliability values of clusters in $\mathbf{\Pi}$ is derived by Eq. (6).
$RDCI$ values of the clusters in the ensemble are achieved according to Eq. (7).
The weighted bipartite graph is constructed using Eq. (10).
METIS algorithm is applied to the weighted bipartite graph to obtain $n^*$ clusters as
  consensus partition
```

Output: Consensus partition is considered to be consensus clustering $\boldsymbol{\pi}^*$.

4.5.3 Reliability Based Hyper Clustering Algorithm

In RBHC algorithm (presented in Algorithm 3) with the idea of merging similar clusters in an approach which is faster than other two methods (it can be experimentally proved later in Section 5.4), the cluster acceptability over other clusters is considered to be pairwise cluster similarity for each cluster. Pairwise similarity of the clusters is the acceptability of a fuzzy cluster over other clusters.

Algorithm 3:: RBHC (Reliability Based Hyper Clustering Algorithm)

```
Input: $\mathbf{\Pi}$, $n^*$, $\boldsymbol{SType}$;
Output: $\boldsymbol{\pi}$*
// $\mathbf{\Pi}$ stands for a pool of fuzzy primary partitions
// $n^*$ stands for number of final clusters
// $\boldsymbol{\pi}^*$ stands for final clustering
// $SType$ determines the membership scheme and could be max, min or sum
Eq. (6) is used to achieve the acceptability of each cluster in $\mathbf{\Pi}$ over the clusters.
k means clusterer algorithm is implemented to extract a partition of the $n^*$ hyper-clusters on
  the primary clusters
The amount that any data point $x_l$ belongs to each hyper-cluster $hc_i$ is calculated as:
  $S\left(\mu_{hc_i}(x_l)\right) = Member\,ship\,Value\,(x_l, hc_i, SType, \boldsymbol{C})$, where $|hc_i|$ is the number of basic clusters
  in hyper-cluster $hc_i$.
The achieved membership degree of each $x_l$ to hyper-cluster $hc_i$ is normalized using Eq. (11),
  i.e., $\mu_{hc_i}(x_l) = \frac{S\left(\mu_{hc_i}(x_l)\right)}{\sum_{j=1}^{n^*} S\left(\mu_{hc_j}(x_l)\right)}$
Output: Consensus clustering $\boldsymbol{\pi}^*$
///////////////////
$Membership\,Value\,(x, hc, SType, \boldsymbol{C})$
switch($SType$)
    case 'max': return $\underset{t\in hc}{\mathrm{Max}} C_t(x)$
    case 'min': return $\underset{t\in hc}{\mathrm{Min}} C_t(x))$
    case 'sum': return $\frac{\sum_{t\in hc} C_t(x)}{|hc|}$
```

Using Eq. (4), acceptability (p) of each cluster is derived over other clusters in the ensemble Π and the pairwise cluster acceptability matrix is obtained which could be clustered into a number of hyper-clusters by the basic clustering algorithms such as k means. After that, depending on the membership scheme (membership value function), the amount that any data point belongs to each hyper-cluster is achieved according to the membership values of the data object to the primary clusters in that hyper-cluster. There are three cases for member scheme. First, the case in which membership scheme is max, membership degree of an arbitrary data object x_l in each hyper-cluster is the maximum membership degrees of the data object in all clusters of the hyper-cluster. Second case is when the membership scheme is min in which the minimum of membership degree of an arbitrary data point x_l to the base clusters in a hyper-cluster is achieved and the minimum achieved value is considered to be membership degree of the data object in that hyper-cluster. The third case is when membership scheme is sum, where the membership degrees of an arbitrary data object x_l in each hyper-cluster is the average of the data object membership degrees in all clusters of the hyper-cluster. Finally, data object membership degrees of each hyper-cluster is divided to sum of the data object membership degrees in all hyper-clusters in order to set this sum to 1. The following notations are used in this paper: hc_i is a hyper cluster, $|hc_i|$ represents the number of base clusters in hc_i, membership degree of x_l in base cluster of C_j is shown by $C_j\left(x_l\right)$, and the membership degree of x_l in hyper-cluster hc_i is shown by $\mu_{hc_i}\left(x_l\right)$ which is obtained according to Eq. (11).

$$\mu_{hc_i}\left(x_l\right)=\frac{S\left(\mu_{hc_i}\left(x_l\right)\right)}{\sum_{j=1}^{c}S\left(\mu_{hc_j}\left(x_l\right)\right)} \tag{11}$$

where $S\left(\mu_{hc_i}\left(x_l\right)\right)$ is the membership degree of the data object x_l in hyper-cluster hc_i which is calculated by scheme function. We can use different basic clustering algorithms depending on their applications to accomplish the clustering task on the primary clusters. As an example, we can replace kmeans clusterer algorithm by Kernelized FCM clusterer algorithm when we need to extract the clusters with different shapes. We can use this algorithm to generate the final fuzzy partition which benefits the smaller computation time in comparison with other consensus algorithms. It will be analyzed in Section 5.4.

5 Experimental Study

5.1 Benchmark

In this paper, several datasets are selected to evaluate the robustness and performance of the proposed fuzzy clustering ensemble approach. Here, for robustness and performance evaluation, we have selected several datasets from UCI machine learning datasets [54], "Galaxy" dataset described in [55] and handmade experimental dataset from a well-known HalfRing dataset as described in Tab. 2.

5.2 Performance Evaluation Criteria

Clustering performance is evaluated using the accuracy (AC) and normalized mutual information (NMI) criteria operating on the crisp clusterings and also the XB and FS criteria operating on the fuzzy clusterings. In the following, a general description on these criteria is provided.

The accuracy measure [28] used for clustering performance evaluation, is one of the most widespread evaluation criteria which creates a sound indication between the final clustering and the ground-truth labels (the prior labeling information) of the examined dataset. When the AC

values are larger, it means that better clustering results are achieved. NMI is another criterion for evaluating clustering performance which is the normalized mutual information between a pair of partitions [30]. Similar to AC metric, a larger NMI value indicates a better clustering result. A fuzzy partition should be converted to a crisp partition before being given to AC and NMI metrics, because these metrics are designed to operate on the crisp partitions. Another criterion for evaluating fuzzy partition π^1 is XB criterion proposed in [47] and modified in [51]. Another criterion is proposed for evaluating fuzzy clustering named FS. A good clustering result obtains a small FS value [51].

Table 2: Specifications of the used datasets

Dataset	Number of data objects (M)	Number of clusters (c)	Number of attributes (N)
Breast	683	2	9
Galaxy	323	7	4
Glass	214	7	10
Haberman	306	2	3
Halfring	400	2	2
Ionesphere	351	2	34
Iris	178	3	13
Knowledge	258	4	5
Seeds	210	3	7
SAHeart	462	2	9
Vowel	990	11	13
Wine	178	3	13
Vehicle	846	4	18
Satimage	6435	7	36
Yeast	1484	8	10

5.3 Base Clustering Generation

By construction of the base clusterings through the FCM clustering algorithm, consensus performance is evaluated over various ensembles. Different numbers of clusters for the FCM method are randomly chosen in the interval $\left[2, \sqrt{M}\right]$; therefore, a diverse set of base clusterings could be made, where M is the data objects number in the under test dataset. The base partitions should be constructed for each dataset before applying all methods to a same set of base partitions. For the considered methods, the performance evaluation, robustness assertion and execution-time are respectively measured for the ensemble size $\beta = 50$, 20 and 10. The proposed approach and the state-of-the-art fuzzy clustering ensembles are assessed using their performance criteria and average robustness criterion (in terms of AC) over 100 runs in the simulation environment (MATLAB) to provide a confident and fair comparison.

5.4 Experimental Results

Achieving a more robust and consistent consensus clustering is the main purpose in a clustering ensemble using combination of several primary partitions. We have compared three proposed consensus algorithms RBFCM, RBGP and RBHC (RBHC contains three versions: (a) RBHC-max, (b) RBHC-sum and (c) RBHC-min) with the state of the art methods, i.e., ITK, sCSPA, sHGPA, sMCLA, sHBGF and the FSCEOGA1 clustering ensemble methods. Performances of the resultant consensus partitions in the proposed and baseline methods are determined by the criteria *AC*, *NMI*, *XB* and *FS*. Note that in each dataset, number of the clusters is the same as the number of them in pre-defined ground truth labels.

Here, each of the proposed methods and the baseline methods are executed 100 times. Tabs. 3–6 respectively exhibit the average values of 100 runs for *AC*, *NMI*, *XB* and *FS* criteria where the bolded value in each row is the best acquired performance of each dataset among all of the algorithms. The average performance of all algorithms is shown in the last column for each dataset, and in the last row, performance of each algorithm averaged over all datasets is shown. Because the FSCEOGA1 and ITK are computationally expensive, these methods cannot handle large datasets because of their large execution time. For this reason (i.e., this method is computationally expensive), the performance results of FSCEOGA1 method are presented on 100 different subsamples on the datasets possessing large numbers of data objects including Satimage, Yeast, Vehicle and Vowel. ITK is also a computationally expensive method which makes us present its performance results on the Satimage and Yeast datasets as averaged on 100 different subsamples.

Table 3: Performance comparison of different algorithms in terms of accuracy

Dataset	sHBGF	sCSPA	sMCLA	ITK	FSCEO-GA1	RBGP	RBFCM	RBHC-max	RBHC-sum	RBHC-min	Dataset avg
Breast	0.8799	0.8389	0.9722	0.8902	0.9508	0.9619	**0.9793**	0.9707	0.9678	0.9707	0.9382
Galaxy	0.3901	0.3406	0.3777	0.4799	0.3808	0.4582	**0.4815**	0.4787	0.4465	0.4358	0.4270
Glass	0.5935	0.6215	0.6075	0.6168	0.5561	**0.6542**	0.6455	0.4871	0.5491	0.5586	0.5890
Haberman	0.7353	0.7353	0.7416	0.7353	0.7153	0.7353	**0.7453**	0.7386	0.7353	0.7451	0.7362
Halfring	0.7500	0.7500	0.7500	0.7800	0.7500	0.7850	0.7775	0.7550	0.7550	**0.9000**	0.7753
Ionesphere	0.6980	**0.7550**	0.6410	0.6410	0.7071	0.7179	0.7251	0.7094	0.7066	0.7236	0.7025
Iris	0.9467	0.9467	0.9133	0.6600	0.8400	0.9533	**0.9700**	0.8293	0.9600	0.8817	0.8901
Knowledge	0.5388	0.5349	0.4767	0.5155	0.5143	**0.6008**	0.5720	0.5853	0.5481	0.5576	0.5444
Seeds	0.9000	0.8143	0.9003	0.5333	0.8295	**0.9005**	**0.9005**	0.8181	0.7998	0.8567	0.8253
SAHeart	0.6537	0.6537	0.6537	0.6537	0.6537	0.6537	**0.6637**	0.6580	0.6565	0.6580	0.6558
Wine	0.9101	0.9045	0.6011	0.8876	0.6944	0.9438	**0.9707**	0.8348	0.8503	0.8003	0.8398
Vehicle	0.4598	0.4090	0.4173	0.4031	0.3917	**0.4693**	0.4663	0.4012	0.4243	0.4062	0.4248
Vowel	0.3556	0.2929	0.1879	0.1949	0.1784	0.3859	**0.4417**	0.1863	0.1891	0.2399	0.2653
Yeast	0.4313	0.4003	0.3854	0.3532	0.3469	**0.4414**	0.4128	0.3865	0.3929	0.4047	0.3955
Satimage	0.7043	0.6194	0.4906	0.5011	0.5129	0.6999	**0.7600**	0.5596	0.6512	0.6120	0.6111
Alg. Avg	0.6631	0.6411	0.6078	0.5897	0.6015	0.6907	0.7008	0.6266	0.6422	0.6501	0.6414

Friedman's ANOVA table

Source	SS	Df	MS	Chi-sq	Prob > chi-sq
Columns	513.133	9	57.0148	57.63	3.82481e-09
Error	688.867	126	5.4672		
Total	1202	149			

Table 4: Performance comparison of different algorithms in terms of ***NMI***

Dataset	sHBGF	sCSPA	sMCLA	ITK	FSCEO-GA1	RBGP	RBFCM	RBHC-max	RBHC-sum	RBHC-min	Dataset Avg
Breast	0.5501	0.4416	**0.8050**	0.5471	0.6977	0.7690	0.7898	0.7979	0.7819	0.7979	0.6978
Galaxy	0.3147	0.2319	0.2678	**0.3612**	0.2309	0.3041	0.3309	0.2465	0.2084	0.1520	0.2648
Glass	0.2707	0.2992	0.3641	0.2979	0.2797	**0.3653**	0.3609	0.2711	0.3490	0.3437	0.3202
Haberman	0.0104	0.0010	0.0516	0.0072	0.0003	0.0104	0.0015	0.0197	0.0143	**0.0627**	0.0179
Halfring	0.2000	0.3000	0.1400	0.3132	0.2886	0.3317	0.3118	0.2986	0.2986	**0.5451**	0.3028
Ionesphere	0.1118	0.2209	0.0000	0.1412	0.1227	0.1591	0.2629	0.1284	0.1468	**0.2733**	0.1567
Iris	0.7980	0.8308	0.8054	0.5453	0.6813	0.8464	**0.8640**	0.7264	**0.8640**	0.7378	0.7699
Knowledge	0.2155	0.2273	0.1017	0.2002	0.2455	**0.2828**	0.2665	0.2806	0.1919	0.2465	0.2259
Seeds	0.6885	0.5264	**0.6972**	0.2349	0.5963	0.6730	0.6852	0.4998	0.4666	0.5999	0.5668
SAHeart	0.0624	0.0665	0.0000	0.0508	0.0741	0.0844	0.0838	0.0910	0.0878	**0.0916**	0.0692
Wine	0.7321	0.7219	0.3866	0.7088	0.4033	0.8051	**0.8466**	0.4256	0.5124	0.5112	0.6054
Vehicle	0.1814	0.1237	0.1472	0.1294	0.1254	0.1943	**0.2015**	0.1093	0.1337	0.1282	0.1474
Vowel	0.3024	0.2512	0.1379	0.1372	0.1341	0.3788	**0.4249**	0.1593	0.1433	0.2194	0.2289
Yeast	0.1320	0.1383	0.1238	0.1109	0.1005	**0.1448**	0.1338	0.1123	0.0991	0.1382	0.1234
Satimage	0.4704	0.4558	0.3440	0.3479	0.3545	0.4889	**0.5534**	0.2186	0.3830	0.3346	0.3951
Alg. Avg	0.3360	0.3224	0.2915	0.2755	0.2890	0.3892	**0.4078**	0.2923	0.3121	0.3455	0.3261

Friedman's ANOVA table

Source	SS	Df	MS	Chi-sq	Prob > chi-sq
Columns	349.83	9	38.8704	38.23	1.58807e-0.5
Error	885.67	126	7.0291		
Total	1235.5	149			

The results of the proposed and other consensus algorithms for different datasets are provided in Tabs. 3–6. From Tab. 3, we can observe that RBGP, RBHC-min and RBFCM outperform other algorithms on five, one and nine datasets respectively while other algorithms do not outperform the rest in any datasets. We can also see that the proposed RBFCM outperforms the other cluster ensemble algorithms in terms of the averaged AC value.

From Tab. 4, we can see that RBGP, RBHC-min, RBFCM outperform the other algorithms on three, four and five datasets. It is also obvious that RBHC-sum, sMCLA and ITK outperform the other algorithms on one, two and one datasets respectively. The averaged performance values on all datasets show that RBFCM algorithm achieves the best performance and RBGP has the next best performance in terms of the averaged NMI value.

Tab. 5 shows that RBHC-sum, RBHC-max and RBHC-min outperform the other algorithms on eight, five and four datasets, respectively while RBFCM outperforms the other algorithms on one dataset. It is also observable that the best averaged XB is achieved by RBHC-max. Also, the proposed RBHC algorithm outperforms the other algorithms on most of the datasets. Using Tab. 6, we can see that, among all algorithms on all datasets, the RBHC methods achieve the best results.

Robustness is a fundamental property for machine learning algorithms, which measures the learning algorithms' tolerance against perturbations (i.e., noises). In order to assess the performance (here, the performance is only accuracy) robustness of the proposed methods as a fundamental property which measures the algorithms' tolerance against perturbations, five infected datasets with 5% to 20% noise ratios are selected from KEEL-dataset repository including Glass, Ionesphere, Iris, SAHeart, and Wine [56]. The datasets are infected with 5% to 20% noise ratios and the ensemble size of this evaluation is $\beta = 20$. Fig. 2 shows the accuracies of the proposed methods against baseline methods of sCSPA, sMCLA, sHBGF and ITK; it is worthy to be mentioned that the accuracies are averaged on all of the infected datasets here. We can conclude from Fig. 2 that RBGP, RBFCM and sHBGF are respectively the most robust methods against the noises in the infected datasets. We can also see that compared with these three methods, RBHC-sum has a lower accuracy but higher robustness.

Table 5: Performance comparison of different algorithms in terms of ***XB***

Dataset	sHBGF	sCSPA	sMCLA	ITK	FSCEO-GA1	RBGP	RBFCM	RBHC-max	RBHC-sum	RBHC-min	Dataset Avg
Breast	3.01	3.16	3.42	2.91	3.62	2.97	2.87	1.76	1.90	**1.78**	2.74
Galaxy	5.64	5.85	5.03	7.91	6.13	3.52	1.54	1.27	**1.18**	1.20	3.93
Glass	2.63	2.97	3.58	3.29	3.19	2.58	0.72	0.76	**0.51**	1.05	2.13
Haberman	5.54	6.07	10.60	5.20	4.62	4.40	3.01	**2.69**	2.75	3.21	4.81
Halfring	4.06	4.06	4.06	20.69	19.46	15.46	**2.02**	2.06	2.76	2.33	7.70
Ionesphere	6.11	4.23	3.88	3.85	6.50	4.95	3.30	2.16	**2.06**	2.07	3.91
Iris	4.52	4.13	3.80	3.20	4.78	3.19	2.00	1.89	2.05	**1.62**	3.12
Knowledge	3.17	3.32	3.90	3.79	2.91	2.68	0.93	**0.83**	0.84	0.87	2.32
Seeds	5.31	6.92	5.47	8.73	7.28	0.43	3.07	**2.78**	3.05	3.00	4.60
SAHeart	2.10	2.07	2.54	2.24	1.98	1.99	1.57	**1.17**	**1.17**	**1.17**	1.80
Wine	1.96	1.86	2.12	2.08	10.88	1.81	1.02	0.69	**0.66**	0.73	2.38
Vehicle	3.94	4.26	4.01	4.50	4.53	3.78	2.13	2.32	**2.12**	2.41	3.40
Vowel	2.22	2.19	2.38	2.14	2.18	1.90	0.42	0.29	**0.27**	0.28	1.43
Yeast	3.88	3.95	4.01	4.02	3.92	3.93	1.19	**0.52**	0.72	**0.52**	2.67
Satimage	2.62	2.62	4.53	3.85	3.47	2.62	2.43	0.96	0.73	0.88	2.47
Alg. Avg	3.78	3.84	4.22	5.23	5.70	3.75	1.88	**1.48**	1.52	1.54	3.29

Friedman's ANOVA table

Source	SS	Df	MS	Chi-sq	Prob > chi-sq
Columns	968.933	9	107.659	106.26	8.47214e-19
Error	262.067	126	2.08		
Total	1231	149			

In Tab. 7, the execution time of the proposed methods is compared with the other related methods for ensemble size of $\beta = 20$ over all of our 15 datasets. The execution time per consensus clustering in all algorithms is computed. It is obvious that among the baseline algorithms, FSCEOGA1, and ITK have longer execution times respectively. Since time complexity is quadratic in terms of number of data objects, co-association based algorithms (sCSPA, RBFCM) are with longer execution times after FSCEOGA and ITK. We can also conclude that considering both the performance and the computational cost, RBHC has better performance than the other algorithms because for most of the datasets, it has a shorter average execution time per consensus with sum membership scheme.

Here, performance assessment of the proposed method is carried out through statistical analysis to assure that the results are not accidental. It is assumed that the data are with normal distribution or homogeneous variance and multiple variables are similar to those in [57] for non-parametric Friedman test [58].

Table 6: Performance comparison of different algorithms in terms of ***FS***

Dataset	sHBGF	sCSPA	sMCLA	ITK	FSCEO-GA1	RBGP	RBFCM	RBHC-max	RBHC-sum	RBHC-min	Dataset Avg
Breast	19735.11	17438.95	23941.22	19553.18	24684.18	18915.31	21289.00	**6592.17**	6955.48	9076.37	16818.10
Galaxy	17779.60	15669.90	19096.80	17913.46	17840.14	15729.97	2480.99	**1750.51**	1880.80	2561.87	11270.40
Glass	17666.25	14088.17	25421.97	14165.89	28888.14	13685.08	4155.32	**1851.03**	2444.11	2799.78	12516.57
Haberman	171490.56	174337.82	207554.16	150037.98	181326.30	171490.56	76916.26	**55867.84**	57232.97	71063.55	131731.80
Halfring	437.44	437.44	437.44	1058.72	1060.59	1059.26	283.36	**285.22**	334.89	308.08	570.24
Ionesphere	30640.53	27886.46	11550.00	23819.98	30678.08	29894.15	23441.92	12370.27	**11942.93**	12547.65	21477.20
Iris	656.13	699.94	734.11	825.03	4435.44	631.90	319.97	**149.52**	289.67	232.75	897.45
Knowledge	423.50	421.81	392.03	396.59	440.36	366.10	107.76	**76.90**	79.64	82.86	278.76
Seeds	16653.57	14436.31	16396.20	9082.09	15047.79	13926.51	6764.35	**3690.49**	5352.22	5507.89	10685.74
SAHeart	10803.54	10878.81	4149.00	11785.28	11093.98	10638.45	8907.31	4439.15	**4221.17**	4373.18	8128.99
Wine	10581.66	10913.89	6553.62	10687.80	113807302.26	11136.85	5615.04	2771.45	**2580.72**	2951.40	11387109.47
Vehicle	257215776.22	217443548.75	356904619.57	242487315.19	234481984.46	224425282.32	131330007.63	**65841206.38**	83276010.24	68782242.16	188218799.30
Vowel	80818.04	77638.84	32294.50	104930.74	115391.37	74901.82	15321.84	6132.49	5973.18	**5582.20**	51898.50
Yeast	133801.46	140489.50	28338.90	28989.12	26203.51	143095.18	40569.40	13008.09	13157.77	**12383.83**	58003.68
Satimage	969282386.39	914618662.68	544472344.13	568545089.13	398880838.19	806486019.98	171743081.84	150451940.90	**119814949.39**	148000752.14	479229606.50
Alg. Avg	81800643.33	75504503.28	60116921.58	54095043.35	49841814.32	68761118.23	20218617.47	14426808.83	**13546893.68**	14460831.05	45277319.51

Friedman's ANOVA table

Source	SS	Df	MS	Chi-sq	Prob > chi-sq
Columns	856.667	9	95.1852	93.64	3.01592e-16
Error	378.333	126	3.0026		
Total	1235	149			

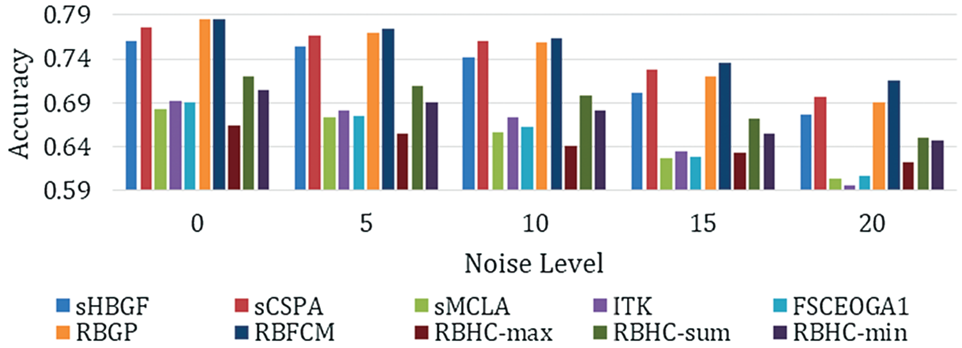

Figure 2: Averaged accuracy of different methods on all of the infected datasets

Table 7: Consumed times of the different algorithms in terms of second

Dataset	sHBGF	sCSPA	sMCLA	ITK	FSCEOGA1	RBGP	RBFCM	RBHC-max	RBHC-sum	RBHC-min
Breast	1.2265	4.5234	0.8768	488.3621	813.8615	1.1016	0.5087	0.0140	**0.0137**	0.0160
Galaxy	0.6780	1.4241	0.5683	89.5773	1684.6240	0.6539	5.0256	**0.0214**	0.0253	0.0293
Glass	0.1707	0.2718	0.1403	35.8495	2428.7338	0.6379	0.3940	0.0317	**0.0235**	0.0346
Haberman	0.4060	0.6921	0.3048	67.3522	448.4177	0.7007	0.1475	0.0178	**0.0152**	0.0169
Halfring	0.7792	1.4650	0.5272	115.9970	313.8434	0.7367	2.3107	0.0194	**0.0175**	0.0207
Ionesphere	0.3985	0.7905	0.2592	129.4893	488.6612	0.6654	0.1032	0.0164	0.0142	**0.0126**
Iris	0.1229	0.1828	0.1317	3.5215	564.0581	0.4078	0.0426	**0.0142**	0.0225	0.0199
Knowledge	0.5598	0.7415	0.4433	64.1924	996.1224	0.5443	0.9201	**0.0143**	0.0159	0.0204
Seeds	0.4029	0.6438	0.3482	31.0934	585.1133	0.4323	0.0559	0.0214	0.0184	**0.0182**
Sheart	0.4570	1.3129	0.2678	242.3952	9193.5573	0.7466	0.6663	0.0307	0.0146	**0.0108**
Wine	0.2298	0.3559	0.2140	16.4309	606.6856	0.3720	0.0599	0.0200	0.0116	**0.0115**
Vehicle	0.7527	4.0906	0.3193	715.7038	Inf	1.1639	2.6166	0.0452	**0.0322**	0.0485
Vowel	0.8877	6.1771	0.3734	4450.8780	Inf	1.2301	47.7147	0.0700	**0.0428**	0.0452
Yeast	1.9942	21.8639	0.9816	Inf	Inf	2.7541	69.6770	0.0754	0.0783	**0.0686**
Satimage	12.7071	922.9457	5.7616	Inf	Inf	18.7358	822.9457	0.2513	**0.1728**	0.3240

Friedman's ANOVA table

Source	SS	Df	MS	Chi-sq	Prob > chi-sq
Columns	1126.83	9	125.204	123.03	3.20515e-22
Error	109.67	126	0.87		
Total	1236.5	149			

Implementing Friedman test [59], we have provided the experimental results in Tabs. 3–7 subject to null hypothesis. The last row of Tab. 3 shows the experimental results according to Friedman test in Tab. 3 where the hypothesis of equal mean rank (in terms of accuracy) is rejected, because of the significant difference shown by probability-value of 3.82E-09. We can also observe that RBFCM, RBGP, sHBGF and RBHC-min are respectively with the highest accuracy scores while ITK has the lowest score. The last rows of Tabs. 4–7 exhibit respectively the Friedman tests of the experimental results for data of Tabs. 4–7. The equal mean rank null hypothesis of the *NMI* (the *XB*, the *FS* and the consumed time) is rejected due to the significant difference provided by probability-value of 1.59E-05 (8.47E−19, 3.02E−16 and 3.21E−22).

Time complexity of RBFCM algorithm is analyzed according to Algorithm 1 which results in complexity of $\mathrm{O}\left(c\left(c+M^2\right)+Mn^*t\right)$, where c is the quantity of all of the fuzzy clusters in the primary partitions, n^* is number of consensus clusters, t represents quantity of the iterations that the FCM needs to converge and M stands for the dataset size. The co-association matrix computation is performed using the term c^2 corresponding to lines 1 to 4 of Algorithm 1. The term cM^2 from line 5 and the term Mn^*t are related to the time complexity of the FCM. For RBGP, the time complexity is acquired as $\mathrm{O}\left(c\left(c+M\right)+Mn^*\beta\right)$ using Algorithm 2, where the terms c^2 and Mc are related to bipartite graph construction corresponding to lines 1 to 4, and line 5 respectively. The term $Mn^*\beta$ is also the METIS time complexity with the ensemble size of corresponding line 6. Algorithm 3 is used for the time complexity calculation of RBHC which is derived as $O\left(c^2+cn^*t+Mn^*\right)$, where Mn^*, c^2 and cn^*t (k means' time complexity) are related to lines 3 and 1 and 2 respectively. In comparison with the other algorithms, RBHC is more efficient in reality, because of the rapid growth of the majority term, i.e., Mn^*, provided that $M \gg c$, and $M \gg n^*$.

6 Conclusions and Future Work

Based on fuzzy cluster unreliability estimation, a novel fuzzy clustering ensemble approach was proposed in this paper. Cluster unreliability was estimated according to an entropic criterion using the cluster labels in the entire ensemble. After that, based on the cluster unreliability and local weighting strategy, a new *RDCI* measure was proposed that is independent on features of the original data and is without any pre-assumption on data distribution. In order to promote the conventional co-association matrix, a local weighting scheme was proposed through the *RDCI* weight (*WFCA*) which determines how much amount each primary fuzzy cluster should contribute in construction of the co-association matrix; indeed, it should be with respect to its reliability in the ensemble. Three algorithms of RBFCM, RBGP and RBHC (suitable for large datasets) were proposed that respectively achieve the final clustering from *WFCA* matrix, the final clustering with respect to the reliability of each cluster and the final clustering by computing the acceptability of each cluster over other clusters of the ensemble. In comparison to other fuzzy clustering ensemble methods, the proposed approach has shown performance improvement and more robustness against noisy datasets through experimental results over fifteen datasets. We concluded that RBHC and RBGP algorithms are appropriate for large datasets because they are linear to the number of data points and have good performance comparison to other fuzzy clustering ensemble methods. It was shown in the paper that RBHC is the best method in terms of time complexity and performance, while RBGP is the best in terms of performance and robustness. RBFCM was also relatively the best in terms of performance. For future works, challenging high dimension datasets in RBFCM and missing values in a dataset can be considered to be problems to be solved. As other challenges for future studies, we can address the diversity impact of the proposed method and its performance in different sampling mechanisms.

Author Contributions: LPY and YXW designed the study and equally contributed the paper as the first author. HG supervised the hypothesis and experiments. LY and YW wrote the manuscript; LY, YW, HP, and KHP edited the manuscript with help from HG and ZM; AB carried out the analyses, implementation of the codes, and the statistical analyses. HP, and KHP generated all figures and tables. All authors have read and approved the final version of the paper.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Maleki, J. E. Contreras-Reyes and M. R. Mahmoudi, "Robust mixture modeling based on two-piece scale mixtures of normal family," *Axioms*, vol. 8, no. 2, pp. 38, 2019.

[2] A. R. Zarei, A. Shabani and M. R. Mahmoudi, "Comparison of the climate indices based on the relationship between yield loss of rain-fed winter wheat and changes of climate indices using GEE model," *Science of the Total Environment*, vol. 661, pp. 711–722, 2019.

[3] M. R. Mahmoudi, M. H. Heydari and Z. Avazzadeh, "On the asymptotic distribution for the periodograms of almost periodically correlated (cyclostationary) processes," *Digital Signal Processing*, vol. 81, pp. 186–197, 2018.

[4] M. Maleki and M. R. Mahmoudi, "Two-piece location-scale distributions based on scale mixtures of normal family," *Communications in Statistics—Theory and Methods*, vol. 46, no. 24, pp. 12356–12369, 2017.

[5] A. R. Nematollahi, A. R. Soltani and M. R. Mahmoudi, "Periodically correlated modeling by means of the periodograms asymptotic distributions," *Statistical Papers*, vol. 58, no. 4, pp. 1267–1278, 2017.

[6] M. Maleki, R. B. Arellano-Valle, D. K. Dey, M. R. Mahmoudi and S. M. J. Jalali, "A Bayesian approach to robust skewed autoregressive processes," *Calcutta Statistical Association Bulletin*, vol. 69, no. 2, pp. 165–182, 2017.

[7] M. R. Mahmoudi, M. Mahmoudi and E. Nahavandi, "Testing the difference between two independent regression models," *Communications in Statistics–Theory and Methods*, vol. 45, no. 21, pp. 6284–6289, 2016.

[8] J. J. Pan, M. R. Mahmoudi, D. Baleanu and M. Maleki, "On comparing and classifying several independent linear and non-linear regression models with symmetric errors," *Symmetry*, vol. 11, no. 6, pp. 820, 2019.

[9] M. R. Mahmoudi, M. H. Heydari and R. Roohi, "A new method to compare the spectral densities of two independent periodically correlated time series," *Mathematics and Computers in Simulation*, vol. 160, pp. 103–110, 2019.

[10] M. R. Mahmoudi, M. H. Heydari and Z. Avazzadeh, "Testing the difference between spectral densities of two independent periodically correlated (cyclostationary) time series models," *Communications in Statistics–Theory and Methods*, vol. 48, no. 9, pp. 2320–2328, 2019.

[11] M. R. Mahmoudi, "On comparing two dependent linear and nonlinear regression models," *Journal of Testing and Evaluation*, vol. 47, no. 1, pp. 449–458, 2018.

[12] M. R. Mahmoudi, M. Maleki and A. Pak, "Testing the equality of two independent regression models," *Communications in Statistics–Theory and Methods*, vol. 47, no. 12, pp. 2919–2926, 2018.

[13] S. B. Rodzman, S. Hasbullah, N. K. Ismail, N. A. Rahman, Z. M. Nor *et al.*, "Fabricated and Shia Malay translated hadith as negative fuzzy logic ranking indicator on Malay information retrieval," *ASM Science Journal*, vol. 13, no. 3, pp. 100–108, 2020.

[14] M. M. Abdulnabi, R. Hassan, R. Hassan, N. E. Othman and A. Yaacob, "A fuzzy-based buffer split algorithm for buffer attack detection in internet of things," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 17, pp. 5625–5634, 2018.

[15] M. A. A. M. Zainuri, E. A. Azari, A. A. Ibrahim, A. Ayob, Y. Yusof *et al.*, "Analysis of adaptive perturb and observe-fuzzy logic control maximum power point tracking for photovoltaic boost DC-DC converter," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 1.6, pp. 201–210, 2019.

[16] Z. M. Rodzi and A. G. Ahmad, "Fuzzy parameterized dual hesitant fuzzy soft sets and its application in TOPSIS," *Mathematics and Statistics*, vol. 8, no. 1, pp. 32–41, 2020.

[17] A. M. S. Bahrin and J. M. Ali, "Hybrid fuzzy-disturbance observer for estimating disturbance in styrene polymerization process," *IOP Conference Series Materials Science and Engineering*, vol. 778, no. 1, pp. 12089, 2020.

[18] B. Choubin, M. Abdolshahnejad, E. Moradi, X. Querol, A. Mosavi *et al.*, "Spatial hazard assessment of the PM10 using machine learning models in Barcelona," *Spain Science of the Total Environment*, vol. 701, pp. 134474, 2020.

[19] E. Dodangeh, B. Choubin, A. N. Eigdir, N. Nabipour, M. Panahi *et al.*, "Integrated machine learning methods with resampling algorithms for flood susceptibility prediction," *Science of the Total Environment*, vol. 705, pp. 135983, 2020.

[20] B. Choubin, A. Mosavi, E. H. Alamdarloo, F. S. Hosseini, S. Shamshirband *et al.*, "Earth fissure hazard prediction using machine learning models," *Environmental Researc*, vol. 179, pp. 108770, 2019.

[21] B. Choubin, M. Borji, A. Mosavi, F. Sajedi-Hosseini, V. P. Singh *et al.*, "Snow avalanche hazard prediction using machine learning methods," *Journal of Hydrology*, vol. 577, pp. 123929, 2019.

[22] S. Shamshirband, E. J. Nodoushan, J. E. Adolf, A. A. Manaf, A. Mosavi *et al.*, "Ensemble models with uncertainty analysis for multi-day ahead forecasting of chlorophyll a concentration in coastal waters," *Engineering Applications of Computational Fluid Mechanics*, vol. 13, no. 1, pp. 91–101, 2019.

[23] S. Qummar, F. G. Khan, S. Shah, A. Khan, S. Shamshirband *et al.*, "A deep learning ensemble approach for diabetic retinopathy detection," *IEEE Access*, vol. 7, pp. 150530–150539, 2019.

[24] S. M. J. Jalali, S. Ahmadian, A. Khosravi, S. Mirjalili, M. R. Mahmoudi *et al.*, "Neuroevolution-based autonomous robot navigation: A comparative study," *Cognitive Systems Research*, vol. 62, pp. 35–43, 2020.

[25] M. R. Mahmoudi, M. Mahmoudi and A. Pak, "On comparing, classifying and clustering several dependent regression models," *Journal of Statistical Computation and Simulation*, vol. 89, no. 12, pp. 2280–2292, 2019.

[26] A. R. Abbasi, M. R. Mahmoudi and Z. Avazzadeh, "Diagnosis and clustering of power transformer winding fault types by cross-correlation and clustering analysis of FRA results," *IET Generation, Transmission & Distribution*, vol. 12, no. 19, pp. 4301–4309, 2018.

[27] S. Golzari, M. N. Zardehsavar, A. Mousavi, M. R. Saybani, A. Khalili *et al.*, "KGSA: a gravitational search algorithm for multimodal optimization based on k-means niching technique and a novel elitism strategy," *Open Mathematics*, vol. 16, no. 1, pp. 1582–1606, 2019.

[28] F. Rashidi, S. Nejatian, H. Parvin and V. Rezaie, "Diversity based cluster weighting in cluster ensemble: An information theory approach," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 1341–1368, 2019.

[29] H. Niu, N. Khozouie, H. Parvin, H. Alinejad-Rokny, A. Beheshti *et al.*, "An ensemble of locally reliable cluster solutions," *Applied Sciences*, vol. 10, no. 5, pp. 1891, 2020.

[30] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, no. 12, pp. 583–617, 2002.

[31] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proc. ICML*, Banff, Canada, pp. 36, 2004.

[32] A. L. N. Fred and A. K. Jain, "Data clustering using evidence accumulation," *Object Recognition Supported by User Interaction for Service Robots*, vol. 4, no. 1, pp. 276–280, 2002.

[33] C. Zhong, X. Yue, Z. Zhang and J. Lei, "A clustering ensemble: Two-level-refined co-association matrix with path-based transformation," *Pattern Recognition*, vol. 48, no. 8, pp. 2699–2709, 2015.

[34] H. Alizadeh, B. Minaei-Bidgoli and H. Parvin, "Optimizing fuzzy cluster ensemble in string representation," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 2, pp. 1350005, 2013.

[35] X. Sevillano, F. Alías and J. C. Socoró, "Positional and confidence voting-based consensus functions for fuzzy cluster ensembles," *Fuzzy Sets and Systems*, vol. 193, no. Supplement C, pp. 1–32, 2012.

[36] J. C. Bezdek, R. Ehrlich and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2–3, pp. 191–203, 1984.

[37] K. Punera and J. Ghosh, "Consensus-based ensembles of soft clustering," *Applied Artificial Intelligence*, vol. 22, no. 7–8, pp. 780–810, 2008.

[38] D. Huang, J. H. Lai and C. D. Wang, "Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis," *Neurocomputing*, vol. 170, no. 11, pp. 240–250, 2016.

[39] Z. Tao, H. Liu, S. Li, Z. Ding and Y. Fu, "Robust spectral ensemble clustering via rank minimization," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 1, pp. 4:1–4:25, 2019.

[40] H. Liu, M. Shao, S. Li and Y. Fu, "Infinite ensemble clustering," *Data Mining and Knowledge Discovery*, vol. 32, no. 2, pp. 385–416, 2018.

[41] J. Wu, Z. Wu, J. Cao, H. Liu, G. Chen *et al.*, "Fuzzy consensus clustering with applications on big data," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1430–1445, 2017.

[42] E. Dimitriadou, A. Weingessel and K. Hornik, "A combination scheme for fuzzy clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 7, pp. 901–912, 2002.

[43] I. S. Dhillon, S. Mallela and R. Kumar, "A divisive information-theoretic feature clustering algorithm for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1265–1287, 2003.

[44] E. Bedalli, E. Mançellari and O. Asilkan, "A heterogeneous cluster ensemble model for improving the stability of fuzzy cluster analysis," *Procedia Computer Science*, vol. 102, no. 8, pp. 129–136, 2016.

[45] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, no. 1–2, pp. 83–97, 1955.

[46] J. V. DeOliveira, A. Szabo and L. N. DeCastro, "Particle swarm clustering in clustering ensembles: Exploiting pruning and alignment free consensus," *Applied Soft Computing*, vol. 55, no. 3, pp. 141–153, 2017.

[47] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.

[48] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics–Theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[49] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[50] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[51] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 370–379, 1995.

[52] S. Vega-Pons and J. Correa-Morris, "A survey of clustering ensemble algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 3, pp. 337–372, 2011.

[53] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

[54] S. C. Odewahn, E. B. Stockwell, R. L. Pennington, R. M. Humphreys and W. A. Zumach, "Automated star/galaxy discrimination with neural networks," *The Astronomical Journal*, vol. 103, no. 1, pp. 318–331, 1992.

[55] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[56] A. F. Ernández, J. L. Uengo and J. D. Errac, "KEEL data-mining software tool: Data set repository," *Integration of Algorithms and Experimental Analysis Framework*, vol. 17, no. 1, pp. 255–287, 2011.

[57] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006.

[58] M. Friedman, “The use of ranks to avoid the assumption of normality implicit in the analysis of variance,” *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[59] R. I. Iman and J. M. Davenport, “Approximations of the critical region of the fbietkan statistic,” *Communications in Statistics-Theory and Methods*, vol. 9, no. 6, pp. 571–595, 1980.