Received: 12 September 2017 | Revised: 4 December 2019 | Accepted: 6 January 2020 | Published on: 26 February 2020

DOI: 10.1002/sam.11451



WILEY

RESEARCH ARTICLE

# Dependability-based cluster weighting in clustering ensemble

**Fatemeh Najafi**[1] | **Hamid Parvin**[2,3] | **Kamal Mirzaie**[1] | **Samad Nejatian**[4,5] | **Vahideh Rezaie**[4,5]

[1]Department of Computer Engineering, Maybod Branch, Islamic Azad University, Maybod, Iran

[2]Department of Computer Engineering, Noorabad Mamasani Branch, Islamic Azad University, Noorabad, Iran

[3]Young Researchers and Elite Clubs, Noorabad Mamasani Branch, Islamic Azad University, Noorabad, Iran

[4]Young Researchers and Elite Clubs, Yasooj Branch, Islamic Azad University, Yasooj, Iran

[5]Department of Electrical Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran

**Correspondence**
Hamid Parvin, Young Researchers and Elite Clubs, Noorabad Mamasani Branch, Islamic Azad University, Noorabad, Fars, Iran.
Email: parvin@iust.ac.ir

**Abstract**

After observing the ensemble success in supervised learning (such as classification), it was extended into unsupervised learning. Therefore, cluster ensemble, which merges multiple basic data partitions or clusters (called as ensemble pool) into an ordinarily better clustering solution usually named as consensus partition, emerged. Any cluster ensemble method tries to optimize a particular criterion during extracting the consensus partition out of the ensemble pool. But traditional cluster ensembles consider all the pool members with the equal importance in making the consensus partition; that is to say that each basic partition or cluster participates in the cluster ensemble algorithm equivalently. Indeed, they ignore to consider any ensemble member according to its importance. But it is obvious that some clusters with more quality deserve more emphasis and some clusters with less quality deserve less emphasis during generating consensus partition. This paper proposes (a) a metric to evaluate quality of any arbitrary cluster, (b) a mechanism to project the computed quality of a cluster into a meaningful weight value, and (c) an approach to apply the weight values of the basic clusters in the cluster ensemble process. Experimental results conducted on a number of real-world standard datasets indicate that the proposed method outperforms the state of the art methods.

**KEYWORDS**

cluster dependability, clustering ensemble, consensus partition, entropy

## 1 | INTRODUCTION

Inspired by emergence of successful ensembles in the supervised learning [4,8,25], clustering/cluster ensemble has emerged. Cluster ensemble merges multiple data clusters (or data partitions) to construct a consensus partition with a better quality [1]. Cluster ensemble generates consensus partition which is more robust, novel, and stable than the output partitions of the simple clustering algorithms [46,47,55].

Two problems are involved in the combinational classification (or the classification ensemble) [23,36]: (a) the problem of generating ensemble pool (or ensemble generation problem), and (b) the problem of extracting the final or consensus classification results (or consensus function problem). The mentioned two mentioned problems are involved in the cluster ensemble, like the combinational classification. In the first problem in the combinational classification, the quality of the base classifiers in the ensemble pool and the diversity among their outputs are

two essential preconditions. Therefore, they can be considered two essential preconditions in a clustering ensemble. But it is even a more challenging task to achieve the mentioned two preconditions in a clustering ensemble due to lack of a clear definition for quality in clustering concept [9].

The ensemble generation problem in the cluster ensemble can use many various approaches for generating the basic partitions of the ensemble pool, such as following ones:

1. Employing various clustering methods [47],
2. Employing an unstable clustering method on different data subspaces [46,47],
3. Employing an unstable clustering method with various initializations [18,46],
4. Employing a same clustering method on various subsampled data [12,15,30,47].

Consensus function problem is the problem that explores among all possible partitions so as to find a partition that has the most similarity with all ensemble members. By the way, the problem of consensus function is known as a NP-complete problem in the cluster ensemble problem [7]. The problem of discovering consensus partition in the cluster ensemble is a more challenging task rather than the problem of discovering consensus classification in the classifier ensemble. An additional challenging task in the cluster ensemble comparing to the combinational classification is to match clusters in different partitions of an ensemble pool.

Two important factors in the ensemble generation problem are (A) cluster quality and (B) between-ensemble diversity. The first factor means that quality of the clusters in the ensemble is important; consequently, it should be considered during the ensemble generation problem. There are three mechanisms to guarantee it: (A.a) to generate an ensemble of high-quality clusters, (A.b) to generate a number of clusters irrespective of their qualities, and then to filter them according to their qualities, and (A.c) to produce a consensus function which extract the consensus partition according to the clusters' weights assigned to clusters according to their qualities. The second factor means that the clusters of the ensemble should be diverse. There are again two approaches to guarantee it: (B.a) to generate an ensemble with a high-level between-ensemble diversity and (B.b) to generate an ensemble of clusters irrespective of their diversity, and then to select a subset of them which is as diverse as possible.

Although the diversity factor in ensemble generation problem has been considered in some works, the other factor has been ignored. Some other works have first generated an ensemble with a high-level between-ensemble diversity; and subsequently they have *assigned* each partition to a weight value. The consensus partition is extracted from the ensemble members according to their weights [20,28,53]. These approaches suffer from (a) eliminating high-quality clusters in bad partitions, and (b) emphasizing the unsuitable clusters in high-quality partitions. The only work that considers the clusters' weighting according to their qualities and ensemble diversity is presented by Zhong et al. [56]. But it also needs the original dataset during ensemble generation and it is also slow. It has also some assumptions about the dataset distribution.

## 2 | RELATED WORK

Cluster ensemble combines multiple data partitions into a single usually better partition often referred to as consensus partition. The consensus partition is usually better in terms of stability, quality, and robustness [42,57].

The quality of a partition can be evaluated by an internal measure or an external measure. If the ground truth labels are used during a partition evaluation measure, then the measure will be considered as an external measure; otherwise, it will be considered as an internal measure. Some examples of internal measures are Sum of Square Error (*SSE*), Calinski Harabasz Index (*CHI*), Davies Bouldin Index (*DBI*), Silhouette Index (*SI*), Dunn Index (*DI*), NIVA Index (*NIVAI*), and so on. Some examples of external measures are Normalized Mutual Information (*NMI*), *F*-Measure (*FM*), Adjacent Rand Index (*ARI*), and Accuracy criterion (*Acc*), and so on. The internal measures are the functions which take a partition and the original data points of dataset as their inputs, and return a partition value as their output. But the external measures are the functions which take a partition and the ground truth labels of dataset as their inputs, and return a partition value as their output. Both types of partition evaluation measures are world-widely accepted. We use external partition evaluation measures to assess quality of resultant clusterings in this paper.

It is worthy to be mentioned that the average quality of partitions generated by a clustering algorithm is considered the quality of that clustering algorithm. The average similarities between partitions generated by a clustering algorithm are considered the stability of that clustering algorithm. To compute the stability of a clustering algorithm, first the clustering algorithm is run $\alpha$ times and $\alpha$ partitions are produced. After that, similarities of all pairs of those partitions are computed, that is, there are $\binom{\alpha}{2}$ pairs of partitions and consequently $\binom{\alpha}{2}$ similarity values. The similarity of a pair of partitions (or two

partitions) is computed as the *NMI* value of those two partitions.

By adding random noise to a given dataset, the partition produced by a clustering algorithm can be changed. If a clustering algorithm is executed on a dataset $\beta$ times and each time a random noise at a same level of $\lambda$ is added to the dataset, then we will have $\beta$ partitions which may be different. The average of all similarity values between all $\binom{\beta}{2}$ pairs of partitions is considered as the clustering algorithm quality at that level noise. A clustering algorithm $A_1$ is considered to be more robust than another clustering algorithm $A_2$, if the quality of the clustering algorithm $A_1$ is degraded less by adding different noise levels to a dataset.

## 2.1 | Ensemble generation problem

In the ensemble generation problem, different approaches can be placed in one of the two general types: (a) clustering ensemble approaches with heterogeneous clustering algorithms and (b) clustering ensemble approaches with homogenous clustering algorithms. The clustering ensemble approaches with heterogeneous clustering algorithms use the different clustering algorithms during generation of the ensemble pool, that is, each partition in the ensemble pool is generated by a different clustering algorithm [46]. The clustering ensemble approaches with homogenous clustering algorithms employ a same clustering algorithm during generation of the ensemble pool, that is, all partitions of the ensemble pool are generated by a same clustering algorithm. The partitions of the ensemble pool in homogenous clustering algorithms can be produced by one of the following subtypes:

1. by employing different initializations of a given clustering algorithm [2,3,40],
2. by employing different parameters (like different numbers of clusters) for data clustering using a same clustering algorithm [3,5],
3. by employing different data projections for data clustering using a same clustering algorithm [12,47],
4. by employing different subsets of dataset features for data clustering using a same clustering algorithm [17],
5. by employing meta heuristic algorithms [24,34,38,39,43,52] for data clustering [42], and
6. by employing different datasets for data clustering using a same clustering algorithm.

The last subtype of the homogenous clustering algorithms, that is, employing different datasets, the different datasets can be generated by one of the following subtypes.

6.1. deterministic subset selection where a number of different subsets of dataset are selected based on a deterministic approach [46],
6.2. nondeterministic subset selection where a number of different subsets of dataset are selected based on a nondeterministic approach [46]. The nondeterministic subset selection can be like bagging [29–31,49] or like boosting [41]. Bagging can be either bootstrap (sampling with replacement) or subsampling (sampling without replacement).

## 2.2 | Consensus function problem

The consensus function problem can be done on either (A) a pool of dendrograms (if the basic clustering algorithms are of hierarchical type) or (B) a pool of partitions. The consensus functions which take a pool of dendrograms as their input [32] are inherently different from the consensus functions of the second category.

The second category contains six subcategories: (1) co-association based approaches, (2) hypergraph partitioning based approaches, (3) median partition based approaches [11], (4) information theoretic based approaches like Quadratic Mutual Information (*QMI*) [47], (5) voting based approaches [30,58], and (6) intermediate space based approaches.

The co-association based approaches, first introduced by Fred and Jain [18], first transform the ensemble pool into a similarity matrix whose $(i, j)$th data entry indicates the number of times the $i$th data point and the $j$th data point are simultaneously in a shared cluster [22,50,54]. Then, by applying a simple clustering algorithm like the hierarchical clustering algorithms, the consensus partition is extracted [29,41].

The hypergraph partitioning based approaches first transform the ensemble pool into a hypergraph, then by a hypergraph clustering algorithm, the consensus partition can be extracted. Cluster-based Similarity Partitioning Algorithm (*CSPA*), Hyper Graph Partitioning Algorithm (*HGPA*), Meta-Clustering Algorithm (MCLA) [46], Spectral graph partitioning algorithm (SPEC) [37], and Bi-Partite Graph Partitioning Algorithm [14] are some cases of hypergraph partitioning based approaches.

The median partition based approaches try to find a (consensus) partition which is mostly acceptable by all partitions of the pool [26]. Combinatorial ensemble approaches are of this type ([10,16,19,45]. The intermediate space based approaches consider the ensemble pool as a new intermediate dataset whose features are partitions and then a new clustering algorithm is applied so as to find the consensus partition. Mixture Model (*MM*) [48]

and Bayesian Ensemble (*BE*) [51] are two samples of this type.

Voting based approaches like voting, weighting voting, selective voting, and selective weighting voting methods have been discussed by Zhou and Tang [58]. They have employed a weighted-voting based on normalized mutual information to combine base partitions. While their work is a weighting mechanism on cluster level, it is inherently different from our work.

## 3 | BACKGROUNDS

To propose weighing co-association matrix, we need to define a meaningful weighing mechanism. In the weighing mechanism, a weigh is assigned to each cluster according to its value. Determining the value of a cluster is more challenging than determining the value of a class as we cannot assess a cluster according to ground truth labels. Therefore, we are needed to define a cluster value according to its dependability. Therefore, we present a list of definitions in the below to show how this process can be done.

**Definition 1.** The undependability of a cluster (denoted by $T$) with regard to a reference set $\Pi$ is denoted by $U_{T,\Pi}$ obtained according to Equation (1).

$$U_{T,\Pi} = \frac{1}{B}\sum_{j=1}^{B} u_{T,\pi^j}, \tag{1}$$

where $T$ is a cluster, $\Pi$ is an ensemble, $\pi^j$ is the $j$th partition in the ensemble, $B$ is the ensemble size, and $u_{T,\pi^j}$ is undependability of the cluster $T$ with regard to the $j$th partition in the ensemble and is calculated based on Equation (2).

$$u_{T,\pi^j} = -\sum_{k=1}^{c^j} \frac{T \cap \pi_k^j}{n} \log_2 \frac{T \cap \pi_k^j}{n}, \tag{2}$$

where $c^j$ is the number of clusters in partition $\pi^j$, $n$ is the size of dataset, and $\pi_k^j$ is the $k$th cluster in the $j$th partition in the ensemble. If all samples in the cluster $T$ belong to exactly one cluster in the partition $\pi^j$, then $u_{T,\pi^j}$ will be zero. Zero is the least output of $u_{T,\pi^j}$. If the samples in the cluster $T$ are distributed equally between all clusters in the partition $\pi^j$, then $u_{T,\pi^j}$ will be its maximum value.

**Definition 2.** The First Dependability Measure (*FDM*) for the cluster $T$ considering the ensemble $\Pi$ is defined according to Equation (3).

$$FDM_{T,\Pi} = e^{-\frac{U_{T,\Pi}}{\alpha}}, \tag{3}$$

where $\alpha$ is a real number which determines the cluster undependability effect.

The *FDM* is presented by Definition 2 in a formal notation. $U_{T,\Pi}$ is always a real positive value; consequently, $FDM_{T,\Pi}$ is always a positive real number less than or equal to one. For all of the clusters in the ensemble of Figure 1A, the $FDM_{T,\Pi}$ values are calculated, and then their *FDM* values are represented in Figure 1B. The variable $\alpha$ has been set to 0.4.

Empirically, for $\alpha \leq 0.2$, the *FDM* falls significantly by increasing the cluster undependability. The choosing values larger than 2 for $\alpha$ results in the linear-correlation between dependability and undependability. The best values for $\alpha$ can be $0.2 \leq \alpha \leq 1$. It will be experimentally shown that 0.4 is a good option for $\alpha$.

**Definition 3.** The Second Dependability Measure (*SDM*) for the cluster $T$ considering the ensemble $\Pi$ is defined according to Equation (4).

$$SDM_{T,\Pi} = 1 - \frac{1}{B}\sum_{j=1}^{B} \dot{u}_{T,\pi^j}, \tag{4}$$

where $B$ is the ensemble size, and $\dot{u}_{T,\pi^j}$ is computed based on Equation (5).

$$\dot{u}_{T,\pi^j} = \frac{u_{T,\pi^j}}{\log_2 c^j}, \tag{5}$$

where $c^j$ is the number of clusters in partition $\pi^j$. It is easy to show that $0 \leq SDM_{T,\Pi} \leq 1$.

The co-association matrix has been introduced in 2005 by Fred and Jain [18]. One of the main drawbacks in co-association matrix based approaches is their inability to consider the weights of clusters. Therefore, we propose cluster weighing co-association matrix in Definitions 4 and 5 so as to deal with this drawback.

**Definition 4.** Given an ensemble $\Pi$, the cluster weighing co-association matrix based on *FDM* or *SDM* can be defined according to Equations (6) and (7).

$$FDMCA_{ij} = \sum_{k=1}^{B}\sum_{q=1}^{c^k} FDM_{\pi_q^k,\Pi} \times (v_{\pi_q^k} v_{\pi_q^k}^T), \tag{6}$$

$$SDMCA_{ij} = \sum_{k=1}^{B}\sum_{q=1}^{c^k} SDM_{\pi_q^k,\Pi} \times \left(v_{\pi_q^k} v_{\pi_q^k}^T\right), \tag{7}$$

where $v_{\pi_q^k}^T$ is transpose of $v_{\pi_q^k}$ and $v_{\pi_q^k}$ is a vector whose $i$th element is defined based on Equation (8).

$$v_{\pi_q^k}(i) = \begin{cases} 1 & i \in \pi_q^k \\ 0 & i \notin \pi_q^k \end{cases}. \tag{8}$$

**Definition 5.** Given an ensemble $\Pi$ and the matrices *FDMCA* and *SDMCA* computed based on Definition 4, the normalized cluster weighing co-association matrix based

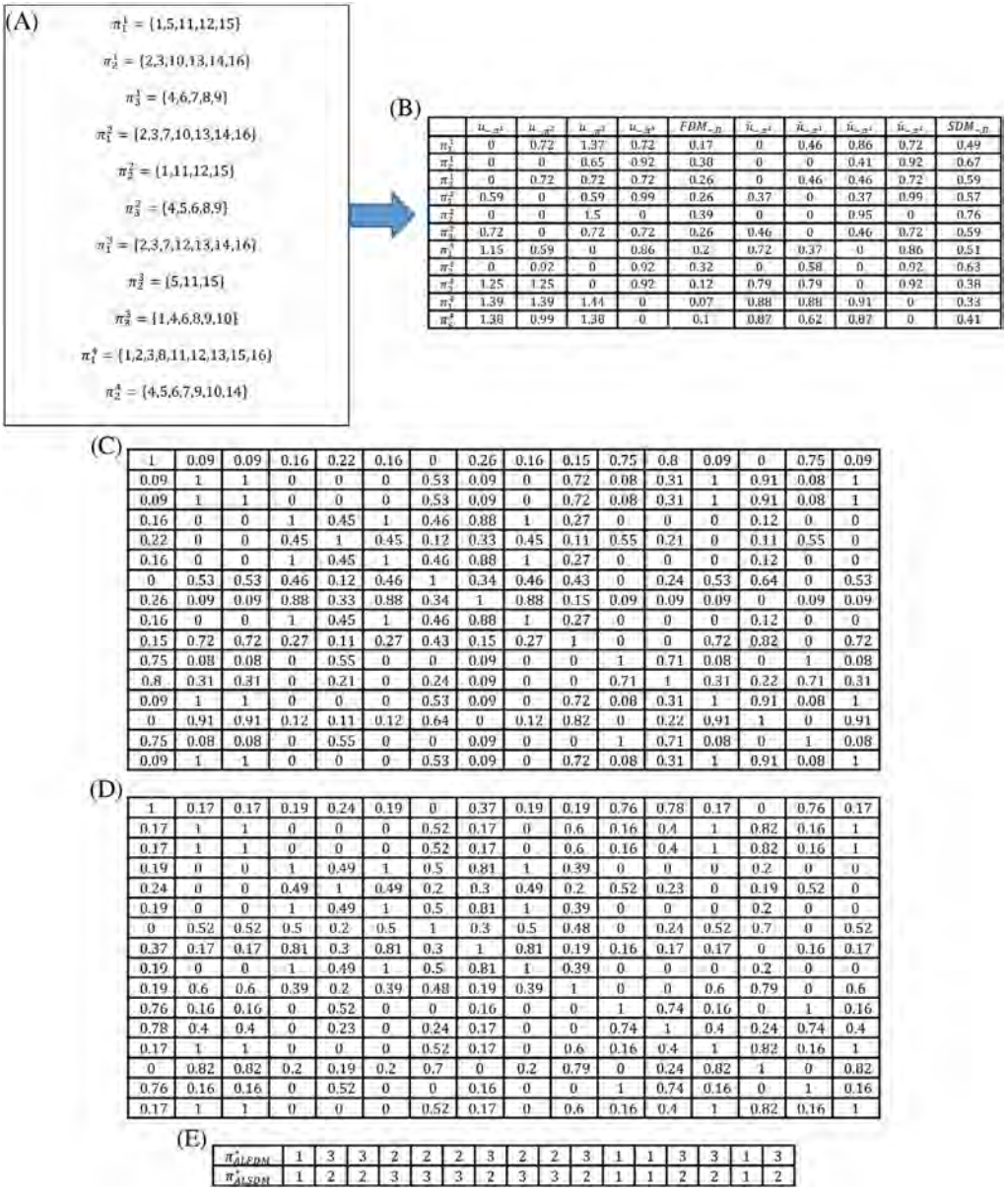



**FIGURE 1** (A) An exemplary ensemble Π with the size of 4, the first three partitions contain three clusters and the last partition has two clusters. (B) Computation of $FDM_{-,\Pi}$ and $SDM_{-,\Pi}$ for all clusters of the ensemble Π. (C) The matrix *NFDMCA* for the ensemble Π. (D) The matrix *NSDMCA* for the ensemble Π. (E) The consensus partitions $\pi^*_{ALFDM}$ and $\pi^*_{ALSDM}$ for the ensemble Π

on *FDM* or *SDM* can be defined according to Equations (9) and (10).

$$NFDMCA_{ij} = \frac{FDMCA_{ij}}{\sqrt{FDMCA_{ii} \times FDMCA_{jj}}}, \tag{9}$$

$$NSDMCA_{ij} = \frac{SDMCA_{ij}}{\sqrt{SDMCA_{ii} \times SDMCA_{jj}}}. \tag{10}$$

In the computation of the matrices *NFDMCA* and *NSDMCA*, we have considered cluster weighting strategy according to cluster dependability. In the new mechanism, the better clusters (clusters with more dependability) participate more during constructing co-association matrix and consequently consensus partition. Through this mechanism, a data point that usually falls into a dependable cluster is highly likely to be placed into a correct cluster. For all clusters in ensemble of Figure 1A, the $u_{T,\pi^j}$, $U_{T,\Pi}$, $\dot{u}_{T,\pi^j}$, $SDM_{T,\Pi}$, and $FDM_{T,\Pi}$ values are calculated in Figure 1B. The matrices *NFDMCA* and *NSDMCA* are depicted in Figure 1C,D respectively.

## 4 | PROPOSED FRAMEWORK

The suggested paradigm for clustering ensemble is focused around an application of the theory of undependability

**FIGURE 2** Pseudocode of the proposed cluster ensemble approach

***input***:
$c$: The quantity of target clusters;
$bca$: The basic clusterer algorithm;
$B$: The ensemble size;
$D$: The given database;
$P_t$: The aimed clustering defined based on real labels;
$\alpha$: One of algorithm parameters;
***output***: $\pi^*_{ALFDM}$; $\pi^*_{ALSDM}$; $nmi_{OutALFDM}$; $nmi_{OutALSDM}$; $acc_{OutALFDM}$; $acc_{OutALSDM}$; $fm_{OutALFDM}$; $fm_{OutALSDM}$; $ar_{OutALFDM}$; $ar_{OutALSDM}$;
01. $n = |D|$;
02. $b = 0$;
03. *For* $i = 1 \ldots B$
03.1. $vc(i) = extract\ a\ random\ integer\ number\ out\ of\ the\ interval\ [2; \sqrt{n}]$;
03.2. $b = b + vc(i)$;
03.3. $\pi^i = bca(D, vc(i))$;
*End*
04. *For* $ii = 1 \ldots B$
04.1. *For* $j = 1 \ldots vc(ii)$
04.1.1. *For* $i = 1 \ldots B$
04.1.1.1. compute $\dot{u}_{\pi^{ii}_j,\pi^i}$; $u_{\pi^{ii}_j,\pi^i}$;
*End*
04.1.2. compute $FDM_{\pi^{ii}_j,\Pi}$; $SDM_{\pi^{ii}_j,\Pi}$;
*End*
*End*
05. compute $FDMCA_{ij}$; \\ Normalize $FDMCA_{ij}$ so that all of its diagonal values are equal to 1.
06. compute $SDMCA_{ij}$; \\ Normalize $SDMCA_{ij}$ so that all of its diagonal values are equal to 1.
07. $\pi^*_{ALFDM} = ALHC(NFDMCA, c)$; \\ $ALHC$ is average linkage hierarchical clustering
08. $\pi^*_{ALSDM} = ALHC(NSDMCA, c)$;
09. $nmiOutALFDM = NMI - M(\pi^*_{ALFDM}, P_t)$;
10. $nmiOutALSDM = NMI - M(\pi^*_{ALSDM}, P_t)$;
11. $accOutALFDM = A - M(\pi^*_{ALFDM}, P_t)$;
12. $accOutALSDM = A - M(\pi^*_{ALSDM}, P_t)$;
11. $fmOutALFDM = F - M(\pi^*_{ALFDM}, P_t)$;
13. $fmOutALSDM = F - M(\pi^*_{ALSDM}, P_t)$;
14. $arrOutALFDM = ARI - M(\pi^*_{ALFDM}, P_t)$;
15. $arrOutALSDM = ARI - M(\pi^*_{ALSDM}, P_t)$;

and weighting of clusters. Figure 2 shows the pseudocode for our approach. In statement 01, the dataset size is initially computed. In statement 03, our base partitions are generated. The $i$th base clustering of the generated ensemble contains $vc(i)$ clusters. Here, $vc(i)$ is a random integer number in interval $[2; \sqrt{n}]$ and $n$ is size of dataset. In each iteration of the statement 03, a random integer number denoted by $vc(i)$ in interval $[2; \sqrt{n}]$ is first produced. Then, the dataset is partitioned into $vc(i)$ clusters through a basic clusterer algorithm denoted by *bca*.

In the statement 04, the *FDM* and *SDM* are calculated for all clusters with regard to the ensemble. The cluster weighing co-association matrices based on *FDM* and *SDM* are computed in the statement 05 and the statement 06 respectively. The consensus partitions $\pi^*_{ALFDM}$ and $\pi^*_{ALSDM}$ are computed respectively in the statement 07 and the statement 08. Hierarchical agglomerative clustering algorithms, that is, average linkage clustering algorithm, have been used as consensus function. The last eight statements are dedicated to evaluating consensus partitions.

Consensus partitions $\pi^*_{ALFDM}$ and $\pi^*_{ALSDM}$ are presented in Figure 1E for the exemplary ensemble presented in Figure 1A for $\alpha = 0.4$.

## 5 | EXPERIMENTATIONS

In the current part, the proposed technique is assessed vs the cutting edge cluster ensemble strategies on a various collection of real datasets. The whole of empirical investigations are accomplished using Matlab2015. The proposed technique is evaluated against a portion of the best strategies in the field such as: Hybrid Bi-Partite Graph Formulation (*HB_PGF*) [14], Sim-Rank Similarity (*SRS*) [21], Weighted-Connected Triple (*W_CT*) [22], Cluster Selection-Evidence Accumulation Clustering (*CS_EAC*) [2], Weighted-Evidence Accumulation Clustering (*W_EAC*) [20], Wisdom of Crowds Ensemble (*WCE*) [3], Graph Partitioning with Multi-Granularity Link Analysis (*GPM_GLA*) [20], and Two_level Co-Association Matrix Ensemble (*TCAME*) [56], Elite Cluster Selection-Evidence Accumulation Clustering (*ECS_EAC*) [40], Cluster-Level Weighting-Graph Clustering (*CLW_GC*) [35], and Robust Clustering Ensemble based on Iterative Fusion of base Clusters (*RCEIFC*) [33]. These techniques utilize the default suggestions of parameters by their relating authors.

### 5.1 | Benchmarks

Throughout all experimentations, 19 real benchmark databases are utilized. The name and detail of each benchmark database are exhibited in Table 1. Saving MNIST [27] and USPS [13], the rest of benchmark databases are from UCI machine learning repository [6]. Earlier performing any experimental test, the whole of benchmark databases are initially standardized with the goal that any attribute in any used benchmark dataset is mapped into interval [0, 1]. It implies that before performing any

**TABLE 1** The employed datasets

| Database title | Number of samples | Number of features: number of target clusters |
|---|---|---|
| Semeion (*S*) | 1593 | 256:10 |
| Multiple-Features (*MF*) | 2000 | 649:10 |
| Image-Segmentation (*IS*) | 2310 | 19:7 |
| Forest-CoverType (*FCT*) | 3780 | 54:7 |
| MNIST (*M*) | 5000 | 784:10 |
| Optical-Digit-Recognition (*ODR*) | 5620 | 64:10 |
| Landsat-Satellite (*LS*) | 6435 | 36:6 |
| ISOLET (*Is*) | 7797 | 617:26 |
| USPS (*U*) | 11 000 | 256:10 |
| Letter-Recognition (*LR*) | 20 000 | 16:26 |
| Breast-Cancer (*BC*) | 683 | 9:2 |
| Bupa (*B*) | 345 | 6:2 |
| Glass (*Gl*) | 214 | 9:6 |
| Galaxy (*Ga*) | 323 | 4:7 |
| SA Heart (*SAH*) | 462 | 9:2 |
| IonoSphere (*IoS*) | 351 | 34:2 |
| Iris (*I*) | 150 | 4:3 |
| Wine (*W*) | 178 | 13:3 |
| Yeast (*Y*) | 1484 | 8:10 |

progression, a preprocessing phase should be taken simply like Equation (21).

$$\ddot{D}_{jk} = \frac{(D_{jk} - \min_{k\in\{1,\dots,D_{1:}\}} D_{jk})}{\max_{k\in\{1,\dots,D_{1:}\}} D_{jk} - \min_{k\in\{1,\dots,D_{1:}\}} D_{jk}}, \quad (21)$$

where $\ddot{D}_{jk}$ is the $k$th standardized attribute in the $j$th object in database $D$, and $|D_{1:}|$ stands for the number of attributes in database $D$.

## 5.2 | Assessment measures

The Normalized Mutual Information Measure (*NMI-M*) [40], *F*-Measure (*F-M*) [40], Accuracy Measure (*A-M*) [40], and Adjacent Rand Index Measure (*ARI-M*) [5] are used for the quality assessment of the consensus partitions obtained by different clustering ensemble approaches. Note that any of these measures takes two parameters (two partitions) as input and returns a real number, which is greater than or equal to zero and less than or equal to one. One of input partition in any of them is usually target partition; the other input partition is a partition generated by a clustering algorithm. Let us assume $P_a$ and $P_b$ are two partitions generated by two clustering algorithm $a$ and $b$. $P_a$ is better than $P_b$, if and only if $X(P_a, P_t) > X(P_b, P_t)$ where $X$ is any of the four mentioned measures, and $P_t$ is the target partition. It is worthy to be mentioned that, all presented results are the mean of 30 distinct runs to make the paper conclusions fairer.

## 5.3 | Tuning algorithm parameters

We use $k$-means clusterer algorithm as the variable *bca* throughout all experimentations. The employed method of selecting seed points is the heuristic initialization of the Kaufman (that has been experimentally proved to be better than stochastic initialization (Pena et al., [44]) and has been required to be followed until explained otherwise). It exactly regenerates a certain fix resulting partition for a given database, each time it is called with a fix $k$. Since we randomly select the value of $k$ from 2 to the square root of the number of data objects, for any run, it should be equal to 141 for the greatest database, equal to 104 for the second biggest database and less than 100 for other databases. Thus, even if a brute force method were implemented rather than stochastically selecting $k$ values, and all $k$ values were selected, we still basically could not get 100 separate resultant clusterings for any database, saving the two aforementioned databases. Therefore, we use stochastic initialization $k$-means to make up an ensemble of size 100 alongside heuristic initialization $k$-means for the datasets with number of data objects lower than 10,000.

The ensemble size variable, that is, $B$, is in the form of $10\times i$ where $i$ is an integer number and $1\le i\le 10$. However, it is considered to be 40 by default, that is, $B = 40$. Variable $\alpha$ is also chosen among the set {0.1,0.2,0.4,0.6,0.8,1.0,2.0,4.0,8.0}. However, it is considered to be 0.4 by default.

It is worthy to be mentioned that, all results are measured in 30 distinct runs. The ensemble members are the same for all approaches for each run. We mentioned the average performances, ultimately.

The variable $\alpha$ seems to have a significant impact on relationship between dependability and undependability. Even a small increase in the degree of undependability leads to a substantial decrease in the level of dependability, for low values of variable $\alpha$. Figure 3 depicts the influence of variable $\alpha$ on *nmiOutALSDM*. As Figure 3 demonstrates, it is suggested that a value should be given to the variable $\alpha$ in the range from 0.3 to 0.7. We then use the value 0.4 for the variable $\alpha$ for the rest of the article.

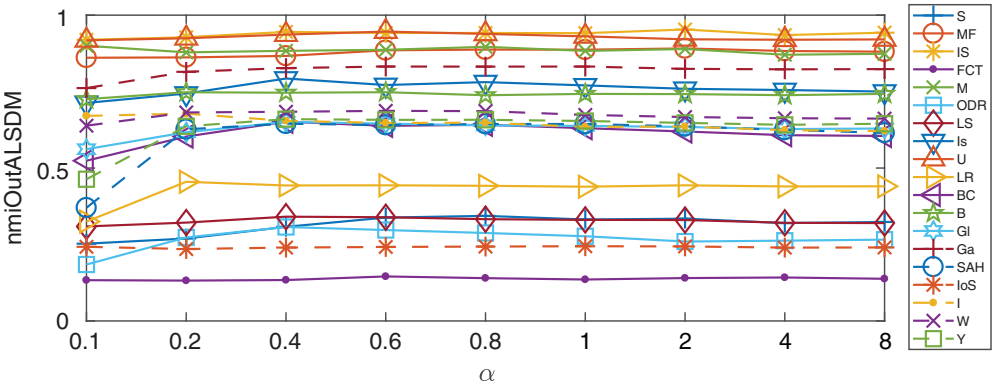



**FIGURE 3** Effect of $\alpha$ on *nmiOutALSDM*

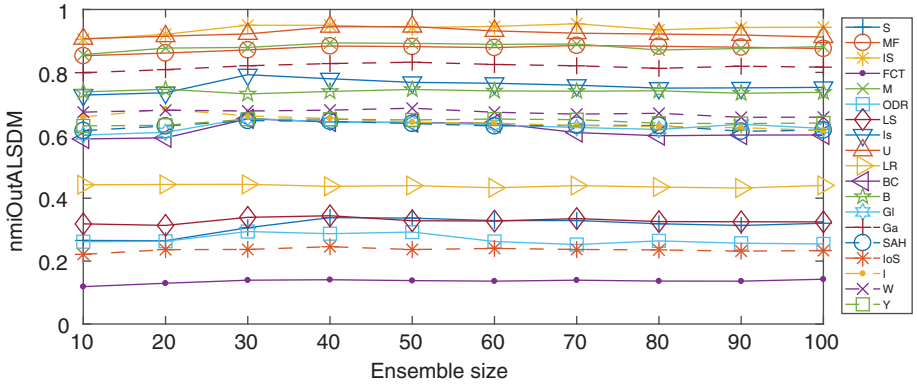



**FIGURE 4** Effect of the ensemble size variable on *nmiOutALSDM*

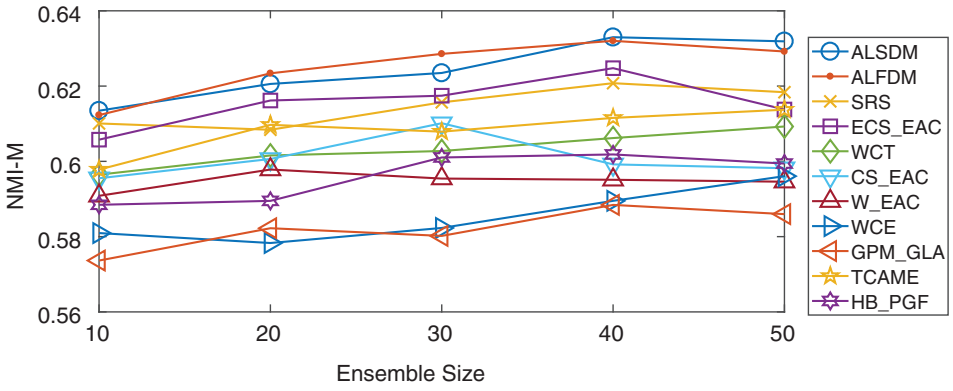



**FIGURE 5** The *NMI-M* of various clustering ensemble approaches in terms of ensemble size; the results are reported as average on all of 19 databases

A number of experiments are performed to determine the best ensemble size for variable $B$, and the performance of our clustering ensemble in terms of different values for $B$ is demonstrated by Figure 4. The influence of ensemble size variable on *nmiOutALSDM* is analyzed in Figure 4. Assigning a value about 40 to ensemble size variable is suggested according to the results presented in Figure 4. Therefore, we employ value 40 for ensemble size variable for the rest of the article, that is, $B = 40$.

In addition, we carried out another analysis to determine the efficacy of our approach and modern approaches. Figure 5 depicts the final results of this analysis. Therefore, throughout 30 distinct runs on a given database, we initially assess an approach in terms of *NMI-M* (it is worth mentioning that in each run the ensemble size is equivalent to $B$), and after that the average output throughout those 30 distinct runs is reported as the quality of the approach on the aforementioned database for ensemble size $B$. After calculating efficiency of an approach on any of our 19 databases with an ensemble size equal to $B$, the average value over all of our 19 databases is assumed to be the efficiency of that approach for ensemble size $B$. After we calculate performance of any ensemble approach in terms of several values of variable $B$, Figure 5 is plotted. Nonetheless, Figure 5 illustrates the efficiency of the different clustering approaches over all databases in terms of *NMI-M*. Figure 6 also describes the efficiency of the different approaches over all databases in terms of *F-M*.

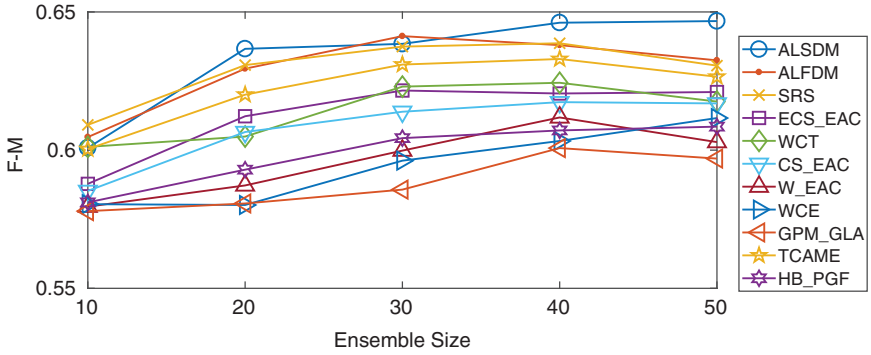



**FIGURE 6** The *F-M* of various clustering ensemble approaches in terms of ensemble size; the results are reported as average on all of 19 databases

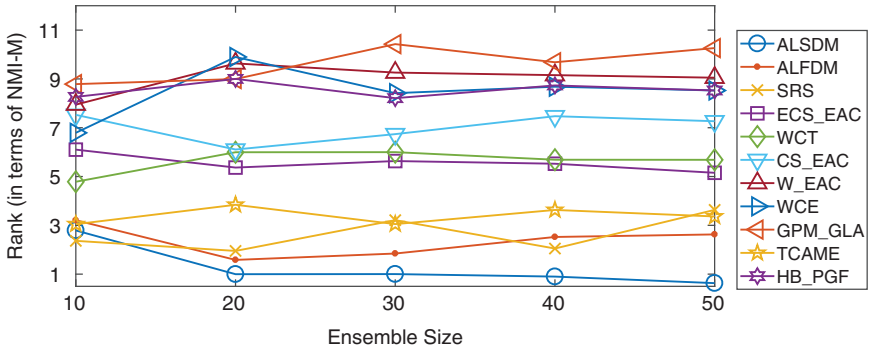



**FIGURE 7** The average rank of the different methods in terms of ensemble size

We need to use some consistent type of *NMI-M* that would make sense when applying a single clustering approach across all databases. Therefore, we use the ranks of different clustering approaches defined based on *NMI-M* instead of their own *NMI-M* values. In a certain database, the approach has the rank of 1 if and only if it provides the maximum *NMI-M* value in that database relative to all other approaches and so on. Average ranks defined based on *NMI-M* values can be more informative than *NMI-M* values. As an instance, an average rank equal to 2.5 indicates that rank of the corresponding approach is the second or third best clustering approach throughout all databases. Figure 7 shows the rank of the different clustering ensemble approaches in terms of different ensemble sizes on all of the 19 databases. The results shown by Figures 5 and 6 show that the suggested clustering ensemble approach outperforms the modern clustering ensemble approaches irrespective of ensemble size. The results presented by Figure 7 also approve those presented by Figures 5 and 6. Figure 7 indicates that *ALSDM* seems to be the most robust among all of the analyzed clustering ensemble approaches. Consequently, usage of second defined weighting mechanism seems to be superior to first defined weighting mechanism.

## 5.4 | Experimental results

A fix ensemble has been employed during obtaining the consensus partition of any method; and accordingly their *NMI-M*, *F-M*, *A-M*, and *ARI-M* values are evaluated. The efficacy of the different clustering ensemble approaches in terms of *NMI-M* is shown in Table 2. To summarize our experimentations, the top two clustering ensemble approaches throughout all approaches have been presented in this subsection. According to the results presented in Table 2, our clustering ensemble approach outperforms all modern clustering ensemble approaches in terms of *NMI-M*.

The results similar to those presented in Table 2 are reported in terms of *FM-M*, *ARI-M*, and *A-M* correspondingly in Tables 3–5. According to the results presented in Table 3, our clustering ensemble approach outperforms all modern clustering ensemble approaches in terms of *F-M*.

Based on the results presented in Table 2–5, the clustering ensemble approach outperforms the modern ones in terms of *NMI-M*, *F-M*, *ARI-M*, and *A-M* respectively. It means the proposed approach is superior to other approaches irrespective of the clustering evaluation metric. It is also worthy to be mentioned that the above conclusion is just on the external clustering evaluation metrics.

Our clustering ensemble approach with *SDM* cluster dependability criterion outperforms all the modern ensemble approaches in approximately all databases in terms of *A-M*, based on the results presented in Table 5. The symbol "+" (or "−") in Table 5 shows that the *ALSDM* results are significantly better (or worse) than those of the compared clustering ensemble approach verified by *t* test with the level of confidence 0.95. The symbol "~" suggests that the *ALSDM* performance is not considerably greater

**TABLE 2** The *NMI-M* of various clustering ensemble approaches on all of 19 databases

| Name | *HB_PGF* | *TCAME* | *GPM_GLA* | *WCE* | *W_EAC* | *CS_EAC* | *W_CT* | *ECS_EAC* | *SRS* | *CLW_GC* | *RCEIFC* | *ALFDM* | *ALSDM* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *S* | 0.6151 | 0.6419 | 0.6054 | 0.6178 | 0.6295 | 0.6251 | 0.6413 | 0.6410 | 0.6520 | 0.6442 | 0.6442 | 0.6515 | 0.**6575** |
| *MF* | 0.6255 | 0.**6903** | 0.6326 | 0.6298 | 0.6191 | 0.6337 | 0.6673 | 0.6503 | 0.6857 | 0.6838 | 0.6726 | 0.6726 | 0.6786 |
| *IS* | 0.6287 | 0.5574 | 0.6205 | 0.5918 | 0.5970 | 0.6077 | 0.5976 | 0.6074 | 0.**6352** | 0.6311 | 0.6263 | 0.6259 | 0.6320 |
| *FCT* | 0.2247 | 0.2173 | 0.1926 | 0.2028 | 0.2319 | 0.2327 | 0.2359 | 0.2422 | 0.2350 | 0.2022 | 0.2416 | 0.2586 | 0.**2603** |
| *M* | 0.6173 | 0.5744 | 0.5775 | 0.6106 | 0.6035 | 0.6259 | 0.5882 | 0.6263 | 0.6174 | 0.6367 | 0.6443 | 0.6509 | 0.**6569** |
| *ODR* | 0.7688 | 0.8150 | 0.8388 | 0.7800 | 0.7722 | 0.7886 | 0.7914 | 0.8067 | 0.8189 | 0.8162 | 0.8096 | 0.8178 | 0.**8263** |
| *LS* | 0.5891 | 0.5202 | 0.5860 | 0.5903 | 0.5872 | 0.5907 | 0.5970 | 0.6148 | 0.6207 | 0.**6439** | 0.6263 | 0.6232 | 0.6295 |
| *Is* | 0.6959 | 0.7227 | 0.7483 | 0.6992 | 0.7014 | 0.7129 | 0.7143 | 0.7399 | 0.7384 | 0.7428 | 0.7451 | 0.7483 | 0.**7554** |
| *U* | 0.5704 | 0.5904 | 0.6243 | 0.5662 | 0.5874 | 0.5516 | 0.6101 | 0.5999 | 0.6228 | 0.6138 | 0.6416 | 0.6348 | 0.**6456** |
| *LR* | 0.4301 | 0.4463 | 0.4193 | 0.4204 | 0.4378 | 0.4249 | 0.4350 | 0.4300 | 0.4426 | 0.4131 | 0.4509 | 0.4578 | 0.**4635** |
| *BC* | 0.8218 | 0.8641 | 0.8137 | 0.8424 | 0.8242 | 0.8620 | 0.8676 | 0.8715 | 0.8681 | 0.8645 | 0.8741 | 0.8677 | 0.**8764** |
| *B* | 0.4907 | 0.5204 | 0.4912 | 0.4986 | 0.5019 | 0.5041 | 0.5097 | 0.5164 | 0.5216 | 0.5213 | 0.5285 | 0.5267 | 0.**5342** |
| *Gl* | 0.3217 | 0.3304 | 0.3131 | 0.3067 | 0.3182 | 0.3167 | 0.3155 | 0.3219 | 0.3103 | 0.3312 | 0.3219 | 0.3446 | 0.**3472** |
| *Ga* | 0.2856 | 0.2867 | 0.2857 | 0.2723 | 0.2822 | 0.2626 | 0.2785 | 0.2786 | 0.2607 | 0.2902 | 0.**2946** | 0.2904 | 0.2929 |
| *SAH* | 0.8038 | 0.8432 | 0.7890 | 0.8161 | 0.8129 | 0.8407 | 0.8429 | 0.8501 | 0.8586 | 0.8666 | 0.8596 | 0.8695 | 0.**8730** |
| *IoS* | 0.1111 | 0.1064 | 0.0996 | 0.0930 | 0.1036 | 0.0946 | 0.1178 | 0.1044 | 0.1156 | 0.1218 | 0.1200 | 0.1233 | 0.**1245** |
| *I* | 0.8123 | 0.8551 | 0.8008 | 0.8287 | 0.8236 | 0.8449 | 0.8533 | 0.8653 | 0.8702 | 0.8697 | 0.8608 | 0.8642 | 0.**8726** |
| *W* | 0.7925 | 0.8251 | 0.7855 | 0.8049 | 0.8014 | 0.8139 | 0.8245 | 0.8137 | 0.8601 | 0.8432 | 0.8504 | 0.8586 | 0.**8666** |
| *Y* | 0.3240 | 0.3322 | 0.3143 | 0.3148 | 0.3203 | 0.3111 | 0.3144 | 0.2829 | 0.3209 | 0.3317 | 0.3339 | 0.3325 | 0.**3352** |

(or weaker) than the performance of the compared clustering ensemble approach verified with confidence level 0.95 by *t* test. The last row is a summary of the results presented by Table 5. The results presented by Table 5 demonstrate that the proposed clustering ensemble approach eclipses the modern clustering ensemble approaches totally. The reported results are verified by paired *t* test. The paired *t* test also shows that, in the experimental results shown in Table 5, the dominance of our clustering ensemble approach could not occur by chance.

All the results shown in Tables 2–5 offer the next two claims:

1. Using the suggested cluster stability as the guideline for weighting the clusters will boost consensus partition quality; since both of our consensus functions are better than the modern clustering ensemble approaches throughout almost all experimentations.
2. *ALSDM* outperforms *ALFDM*.

## 5.5 | Robustness analysis

For a database $D$, ${}^{+\rho}D$ is the same dataset with $\rho$ percent added noisy data objects uniformly scattered in feature space, that is, ${}^{+\rho}D = D \cup X_{1:\rho\times|D|}$ and $\rho = 0.05 \times i$ where $i$ is an integer greater than or equal to 0 and less than or equal to 9. Figure 8 shows the *NMI-M* on all of the 19 aforementioned databases in terms of noise level. The results shown in Figure 8 are average over all of the 19 datasets. As shown in Figure 8, the proposed clustering ensemble approach is more robust than the modern clustering ensemble approaches.

## 5.6 | Complexity analysis

The time taken by various clustering ensemble approaches in terms of database size is estimated and afterwards represented in Figure 9. Even though the proposed approach is not the best in computational context, it is still computationally better than several modern approaches to clustering ensemble.

# 6 | CONCLUSIONS AND FUTURE WORK

The article introduces a clustering ensemble approach through using the approximated dependability of clusters

**TABLE 3** The *F-M* of various clustering ensemble approaches on all of 19 databases

| Name | *HB_PGF* | *TCAME* | *GPM_GLA* | *WCE* | *W_EAC* | *CS_EAC* | *W_CT* | *ECS_EAC* | *SRS* | *ALFDM* | *ALSDM* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *S* | 0.6475 | 0.6757 | 0.6373 | 0.6503 | 0.6626 | 0.6580 | 0.6751 | 0.6747 | 0.6863 | 0.6858 | 0.**6921** |
| *MF* | 0.6584 | 0.**7256** | 0.6659 | 0.6629 | 0.6517 | 0.6671 | 0.7024 | 0.6845 | 0.7218 | 0.6972 | 0.7038 |
| *IS* | 0.6618 | 0.6120 | 0.6532 | 0.6229 | 0.6284 | 0.6397 | 0.6290 | 0.6394 | 0.**7213** | 0.6899 | 0.6968 |
| *FCT* | 0.2581 | 0.2595 | 0.2554 | 0.2450 | 0.2546 | 0.2555 | 0.2683 | 0.2744 | 0.2674 | 0.3021 | 0.**3043** |
| *M* | 0.6493 | 0.6178 | 0.6189 | 0.6427 | 0.6353 | 0.6588 | 0.6213 | 0.6598 | 0.6415 | 0.6855 | 0.**6915** |
| *ODR* | 0.8093 | 0.8579 | 0.8793 | 0.8211 | 0.8128 | 0.8301 | 0.8331 | 0.8492 | 0.8720 | 0.8814 | 0.**8898** |
| *LS* | 0.6501 | 0.6228 | 0.6468 | 0.6614 | 0.6481 | 0.6618 | 0.6684 | 0.6872 | 0.6934 | 0.6961 | 0.**7026** |
| *Is* | 0.7325 | 0.7607 | 0.7845 | 0.7360 | 0.7383 | 0.7504 | 0.7519 | 0.7788 | 0.7773 | 0.7879 | 0.**7952** |
| *U* | 0.6320 | 0.6531 | 0.6761 | 0.6276 | 0.6499 | 0.6333 | 0.6738 | 0.6630 | 0.6977 | 0.7027 | 0.**7096** |
| *LR* | 0.4527 | 0.4698 | 0.4414 | 0.4425 | 0.4608 | 0.4473 | 0.4579 | 0.4526 | 0.4659 | 0.4836 | 0.**4879** |
| *BC* | 0.9106 | 0.9574 | 0.9016 | 0.9334 | 0.9132 | 0.9551 | 0.9613 | 0.9656 | 0.9619 | 0.9615 | 0.**9711** |
| *B* | 0.7379 | 0.7825 | 0.7386 | 0.7498 | 0.7548 | 0.7581 | 0.7664 | 0.7766 | 0.7844 | 0.7958 | 0.**8033** |
| *Gl* | 0.3386 | 0.3478 | 0.3296 | 0.3228 | 0.3349 | 0.3334 | 0.3321 | 0.3388 | 0.3266 | 0.3618 | 0.**3655** |
| *Ga* | 0.3006 | 0.3018 | 0.3007 | 0.2866 | 0.2971 | 0.2764 | 0.2932 | 0.2933 | 0.2744 | 0.3055 | 0.**3083** |
| *SAH* | 0.8461 | 0.8876 | 0.8305 | 0.8590 | 0.8557 | 0.8849 | 0.8873 | 0.8948 | 0.9038 | 0.9097 | 0.**9189** |
| *IoS* | 0.1670 | 0.1600 | 0.1498 | 0.1399 | 0.1558 | 0.1422 | 0.1772 | 0.1570 | 0.1739 | 0.1853 | 0.**1872** |
| *I* | 0.8972 | 0.9422 | 0.8850 | 0.9144 | 0.9091 | 0.9315 | 0.9403 | 0.9529 | 0.9581 | 0.9517 | 0.**9606** |
| *W* | 0.8342 | 0.8685 | 0.8268 | 0.8473 | 0.8436 | 0.8567 | 0.8679 | 0.8565 | 0.9054 | 0.9035 | 0.**9122** |
| *Y* | 0.3411 | 0.3497 | 0.3308 | 0.3314 | 0.3372 | 0.3275 | 0.3309 | 0.2978 | 0.3378 | 0.3494 | 0.**3528** |

**TABLE 4** The *ARI-M* of various clustering ensemble approaches on all of 19 databases

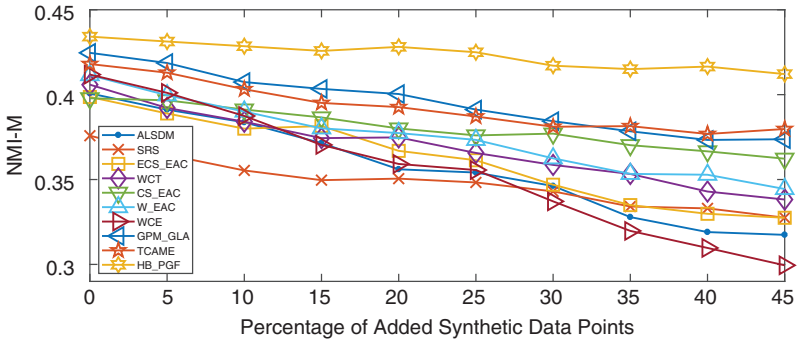
| Name | *HB_PGF* | *TCAME* | *GPM_GLA* | *WCE* | *W_EAC* | *CS_EAC* | *W_CT* | *ECS_EAC* | *SRS* | *ALFDM* | *ALSDM* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *S* | 0.4995 | 0.4003 | 0.5006 | 0.5056 | 0.5026 | 0.5078 | 0.5094 | 0.5174 | 0.4724 | 0.5101 | 0.**5143** |
| *MF* | 0.5050 | 0.**5574** | 0.5217 | 0.5136 | 0.4634 | 0.4913 | 0.4712 | 0.5126 | 0.5259 | 0.5389 | 0.5446 |
| *IS* | 0.5188 | 0.3076 | 0.5430 | 0.5241 | 0.5159 | 0.**5451** | 0.5318 | 0.5494 | 0.5234 | 0.5373 | 0.5423 |
| *FCT* | 0.1139 | 0.1378 | 0.1285 | 0.1367 | 0.1358 | 0.1266 | 0.**1476** | 0.1332 | 0.1234 | 0.1458 | 0.1469 |
| *M* | 0.4889 | 0.3972 | 0.5432 | 0.5506 | 0.4889 | 0.5305 | 0.4995 | 0.5581 | 0.5363 | 0.5646 | 0.**5695** |
| *ODR* | 0.7604 | 0.7106 | 0.7771 | 0.7785 | 0.7568 | 0.7612 | 0.7588 | 0.7755 | 0.7416 | 0.7862 | 0.**7935** |
| *LS* | 0.5321 | 0.5604 | 0.5297 | 0.5362 | 0.5309 | 0.5382 | 0.5436 | 0.5575 | 0.5684 | 0.5681 | 0.**5739** |
| *Is* | 0.5586 | 0.3916 | 0.5544 | 0.5571 | 0.5501 | 0.5468 | 0.5658 | 0.5757 | 0.5661 | 0.5778 | 0.**5833** |
| *U* | 0.4421 | 0.4413 | 0.4501 | 0.4537 | 0.4459 | 0.4512 | 0.4504 | 0.4699 | 0.5051 | 0.5018 | 0.**5069** |
| *LR* | 0.1494 | 0.1249 | 0.1542 | 0.1474 | 0.1589 | 0.1589 | 0.1417 | 0.1643 | 0.1648 | 0.1767 | 0.**1782** |
| *BC* | 0.5443 | 0.5819 | 0.5363 | 0.5671 | 0.5435 | 0.5838 | 0.5907 | 0.5918 | 0.5851 | 0.5925 | 0.**5978** |
| *B* | 0.2447 | 0.2734 | 0.2458 | 0.2521 | 0.2615 | 0.2582 | 0.2604 | 0.2702 | 0.2713 | 0.2874 | 0.**2896** |
| *Gl* | 0.0988 | 0.0882 | 0.0917 | 0.0833 | 0.0935 | 0.0894 | 0.0851 | 0.1067 | 0.0836 | 0.1197 | 0.**1207** |
| *Ga* | 0.0813 | 0.0607 | 0.0775 | 0.0677 | 0.0791 | 0.0646 | 0.0744 | 0.0774 | 0.0595 | 0.0860 | 0.**0862** |
| *SAH* | 0.5253 | 0.5669 | 0.5191 | 0.5348 | 0.5363 | 0.5575 | 0.5606 | 0.5669 | 0.5739 | 0.5863 | 0.**5916** |
| *IoS* | 0.1095 | 0.0961 | 0.0988 | 0.0911 | 0.0938 | 0.0865 | 0.1109 | 0.0952 | 0.1099 | 0.1112 | 0.**1123** |
| *I* | 0.6409 | 0.6789 | 0.6287 | 0.6508 | 0.6481 | 0.6621 | 0.6714 | 0.6888 | 0.6854 | 0.6827 | 0.**6896** |
| *W* | 0.6076 | 0.6399 | 0.5979 | 0.6192 | 0.6117 | 0.6264 | 0.6416 | 0.6245 | 0.6679 | 0.6666 | 0.**6731** |
| *Y* | 0.1427 | 0.1139 | 0.1355 | 0.1424 | 0.1393 | 0.1383 | 0.1422 | 0.1449 | 0.1446 | 0.1525 | 0.**1533** |

**TABLE 5** The *A-M* of various clustering ensemble approaches on all of 19 databases

| Name | *HB_PGF* | *TCAME* | *GPM_GLA* | *WCE* | *W_EAC* | *CS_EAC* | *W_CT* | *ECS_EAC* | *SRS* | *CLW_GC* | *RCEIFC* | *ALFDM* | *ALSDM* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *S* | 0.7029+ | 0.7231+ | 0.6958+ | 0.7049+ | 0.7136+ | 0.7103+ | 0.7226+ | 0.7223+ | 0.7307+ | 0.7013+ | 0.7012+ | 0.7282 | **0.7354** |
| *MF* | 0.7106+ | **0.7589**- | 0.7161+ | 0.7138+ | 0.7059+ | 0.7168+ | 0.7426~ | 0.7294+ | 0.7571- | 0.7424~ | 0.7386+ | 0.7366 | 0.7436 |
| *IS* | 0.7361+ | 0.6549+ | 0.7369~ | 0.6858+ | 0.6896+ | 0.6975+ | 0.6900+ | 0.6973+ | **0.7568**- | 0.7402+ | 0.7382+ | 0.7414 | 0.7485 |
| *FCT* | 0.4814+ | 0.4744+ | 0.4549+ | 0.4702+ | 0.4903+ | 0.4917+ | 0.4996+ | 0.5097+ | 0.4911+ | 0.4931+ | 0.5216+ | 0.5296 | **0.5342** |
| *M* | 0.7272+ | 0.6673+ | 0.6694+ | 0.7096+ | 0.6944+ | 0.7309+ | 0.6727+ | 0.7388+ | 0.7172+ | 0.7482+ | 0.7333+ | 0.7570 | **0.7645** |
| *ODR* | 0.8264+ | 0.8675+ | **0.8864**- | 0.8362+ | 0.8293+ | 0.8437+ | 0.8463+ | 0.8569+ | 0.8711+ | 0.8583+ | 0.8568+ | 0.8694 | 0.8779 |
| *LS* | 0.6839+ | 0.6367+ | 0.6817+ | 0.6948+ | 0.6826+ | 0.6851+ | 0.6896+ | 0.7027+ | 0.7171+ | 0.7148+ | 0.7163+ | 0.7178 | **0.7246** |
| *Is* | 0.7653+ | 0.7872+ | 0.8092+ | 0.7679+ | 0.7707+ | 0.7791+ | 0.7803+ | 0.8016+ | 0.8004+ | 0.8065+ | 0.8080+ | 0.8074 | **0.8148** |
| *U* | 0.6921+ | 0.7069+ | 0.7254+ | 0.6891+ | 0.7046+ | 0.6733+ | 0.7117+ | 0.7039+ | 0.7191+ | 0.7149+ | 0.7166+ | 0.7293 | **0.7359** |
| *LR* | 0.5785+ | 0.5885+ | 0.5717+ | 0.5726+ | 0.5832+ | 0.5754+ | 0.5815+ | 0.5785+ | 0.5862+ | 0.5731+ | 0.5509+ | 0.5917 | **0.5972** |
| *BC* | 0.9145+ | 0.9583+ | 0.9063+ | 0.9356+ | 0.9169+ | 0.9561+ | 0.9621+ | 0.9662+ | 0.9626+ | 0.9645+ | **0.9733**~ | 0.9618 | 0.9715 |
| *B* | 0.7694+ | 0.8045+ | 0.7699+ | 0.7786+ | 0.7825+ | 0.7851+ | 0.7917+ | 0.7998+ | 0.8061+ | 0.8146+ | 0.8127+ | 0.8132 | **0.8214** |
| *Gl* | 0.5161+ | 0.5209+ | 0.5115+ | 0.5087+ | 0.5142+ | 0.5135+ | 0.5128+ | 0.5162+ | 0.5157+ | **0.5312**~ | 0.5235+ | 0.5251 | 0.5302 |
| *Ga* | 0.4969~ | 0.4975~ | 0.4969~ | 0.4895+ | 0.4951~ | 0.4854+ | 0.4932+ | 0.4933+ | 0.4847+ | 0.4943+ | 0.4946+ | 0.4965 | **0.5007** |
| *SAH* | 0.8574+ | 0.8937+ | 0.8441+ | 0.8685+ | 0.8656+ | 0.8913+ | 0.8934+ | 0.9001+ | 0.9083+ | 0.9158+ | 0.9196+ | 0.9131 | **0.9221** |
| *IoS* | 0.4347+ | 0.4317+ | 0.4273+ | 0.4231+ | 0.4299+ | 0.4241+ | 0.4392~ | 0.4304+ | 0.4378~ | 0.4391~ | 0.4305+ | 0.4399 | **0.4436** |
| *I* | 0.9023+ | 0.9438+ | 0.8914+ | 0.9187+ | 0.9131+ | 0.9338+ | 0.9418+ | 0.9542+ | 0.9594~ | **0.9866**- | 0.9846- | 0.9519 | 0.9614 |
| *W* | 0.8472+ | 0.8768+ | 0.8407+ | 0.8584+ | 0.8552+ | 0.8665+ | 0.8763+ | 0.8663+ | 0.9097+ | 0.9036+ | 0.9015+ | 0.9074 | **0.9159** |
| *Y* | 0.5174~ | 0.5219~ | 0.5121+ | 0.5124+ | 0.5154+ | 0.5104+ | 0.5122+ | 0.4955+ | 0.5157+ | 0.5117+ | 0.5141+ | 0.5185 | **0.5235** |
| *Summery* | 17/2/0 | 16/2/1 | 16/2/1 | 19/0/0 | 18/1/0 | 19/0/0 | 17/2/0 | 19/0/0 | 15/2/2 | 15/3/1 | 17/1/1 | — | — |

*Note*: The symbol "+" (or "−") stands for the superiority (inferiority) of the $\pi^*_{ALSDM}$ quality to that of the resultant clustering of a clustering ensemble approach verified by paired *t* test with the 0.95 level of confidence. The symbol "~" shows that the $\pi^*_{ALSDM}$ quality is not significantly superiority (inferiority) to that of the resultant clustering of a clustering ensemble approach verified by paired *t* test with the 0.95 level of confidence.

**FIGURE 8** The noise influence on performance of different clustering ensembles in terms of *NMI-M*




to achieve consensus partition. The article recommends a clustering ensemble approach that includes a few steps. Each cluster's dependability is initially calculated through an entropy calculation and an exponential transform, which represents amount of the cluster's distribution across various clusters of a partition in a reference set. The dependabilities of the various clusters are used during consensus partition generation. We propose an approach that is able to participate any cluster according to its dependability. It is based on weighted evidence accumulation clustering. After that, the proposed solution to clustering ensemble is assessed on 19 real-world databases. The empirical assessment shows that the proposed solution performs better than the existing approaches; therefore, large studies on real-world databases indicate that the proposed approach can be on par with or outperform the modern approaches.

An article contribution is the implementation of the idea of dependability into weighting the cluster ensemble. It is worthy to be mentioned that dependability is indeed

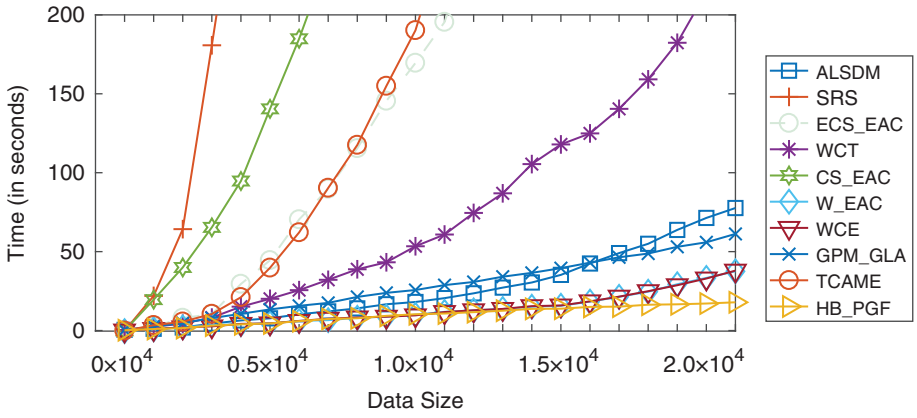



**FIGURE 9** The spent time in terms of number of records in a dataset

a variant of the uncertainty. It is empirically shown that the proposed clustering ensemble approach outperforms the modern clustering ensemble approaches in terms of performance measures, time complexity, and robustness.

The suggested dependability metric suffers from cluster size sensitivity where it punishes any cluster with a greater size. It is the major shortcoming of the suggested dependability metric. Therefore, we have suggested a separate dependability metric to address this drawback. This new metric is less sensitive to the scale of the clusters. As a guideline to the future work, effect of subsampling mechanisms on the proposed clustering ensemble can be evaluated. Another guideline to the future work can be modification of dependability metric so as to become insensitive to cluster size.

## ORCID

*Hamid Parvin* https://orcid.org/0000-0002-5810-1766

## REFERENCES

1. S. Abbasi et al., *Clustering ensemble selection considering quality and diversity*, Artif. Intell. Rev. 52(2) (2019), 1311–1340.
2. H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin, *To improve the quality of cluster ensembles by selecting a subset of base clusters*, J. Exp. Theoret. Artif. Intell. 26 (2014), 127–150.
3. H. Alizadeh, M. Yousefnezhad, and B. Minaei-Bidgoli, *Wisdom of Crowds cluster ensemble*, Intell. Data Anal. 19 (2015), 485–503.
4. S. Amini et al., *Object-based classification of hyperspectral data using Random Forest algorithm*, Geo-spatial Inf. Sci. 21(2) (2018), 127–138.
5. J. Azimi and X. Fern, *Adaptive cluster ensemble selection*, in *Proc. of IJCAI*, 2009, 992–997.
6. K. Bache and M. Lichman, *UCI machine learning repository*, 2013, available at http://archive.ics.uci.edu/ml
7. J. P. Barthelemy and B. Leclerc, *The median procedure for partition*, in *Partitioning Data Sets, AMS DIMACS Series in Discrete Mathematics*, Vol 19, I. J. Cox et al., Eds., 1995, 3–34.
8. O. Brovkina et al., *Unmanned aerial vehicles (UAV) for assessment of qualitative classification of Norway spruce in temperate forest stands*, Geo-spatial Inf. Sci. 21(1) (2018), 12–20.
9. Z. Chang, J. Cao, and Y. Zhang, *A novel image segmentation approach for wood plate surface defect classification through convex optimization*, J. For. Res. 29(6) (2018), 1789–1795.
10. I. T. Christou, *Coordination of cluster ensembles via exact methods*, IEEE Trans. Pattern Anal. Mach. Intell. 33(2) (2011), 279–293.
11. D. Cristofor and D. Simovici, *Finding median partitions using information-theoretical-based genetic algorithms*, J. Univ. Comput. Sci. 8 (2002), 153–172.
12. S. Dudoit and J. Fridlyand, *Bagging to improve the accuracy of a clustering procedure*, Bioinformatics 19(9) (2003), 1090–1099.
13. D. Dueck, Affinity propagation: Clustering data by passing messages, PhD dissertation, University of Toronto, 2009.
14. X. Z. Fern and C. E. Brodley, *Solving cluster ensemble problems by bi-partite graph partitioning*, in *Proc. of International Conference on Machine Learning (ICML)*, 2004.
15. B. Fischer and J. M. Buhmann, *Path-based clustering for grouping of smooth curves and texture segmentation*, IEEE Trans. PAMI 25(4) (2003), 513–518.
16. L. Franek and X. Jiang, *Ensemble clustering by means of clustering embedding in vector spaces*, Pattern Recogn. 47 (2014), 833–842.
17. A.L.N. Fred and A.K. Jain, Data clustering using evidence accumulation, Proc. of the 16th Intl. Conf. on Pattern Recognition, ICPR 2002, Quebec City, 2002, pp. 276–280.
18. A. L. N. Fred and A. K. Jain, *Combining multiple clusterings using evidence accumulation*, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005), 835–850.
19. A. Guénoche, *Consensus of partitions: A constructive approach*, Adv. Data Anal. Classif. 5 (2011), 215–229.
20. D. Huang, J. H. Lai, and C. D. Wang, *Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis*, Neurocomputing 170 (2015), 240–250.
21. N. Iam-On, T. Boongoen, and S. Garrett, *Refining pairwise similarity matrix for cluster ensemble problem with cluster relations*, Proc. of International Conference on Discovery Science (ICDS), 2008, pp. 222–233.
22. N. Iam-On et al., *A link-based approach to the cluster ensemble problem*, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011), 2396–2409.
23. H. Jamalinia et al., *Diverse classifier ensemble creation based on heuristic dataset modification*, J. Appl. Stat. 45(7) (2018), 1209–1226.

24. M. M. Jenghara, H. Ebrahimpour-Komleh, and H. Parvin, *Dynamic protein–protein interaction networks construction using firefly algorithm*, Pattern. Anal. Applic. 21(4) (2018), 1067–1081.
25. M. M. Jenghara et al., *Imputing missing value through ensemble concept based on statistical measures*, Knowl. Inf. Syst. 56(1) (2018), 123–139.
26. R. Jiamthapthaksin, C.F. Eick, S. Lee, *GAC-GEO: A generic agglomerative clustering framework for geo-referenced datasets*, Knowl. Inf. Syst., (2010).
27. Y. LeCun et al., *Gradient-based learning applied to document recognition*, Proc. IEEE 86 (1998), 2278–2324.
28. T. Li and C. Ding, *Weighted consensus clustering*, Proc. of SIAM International Conference on Data Mining (SDM), 2008.
29. B. Minaei-Bidgoli et al., *Effects of resampling method and adaptation on clustering ensemble efficacy*, in *Artificial Intelligence Review*, Springer, 2014. https://doi.org/10.1007/s10462-011-9295-x.
30. B. Minaei-Bidgoli, A. Topchy, W.F. Punch, *Ensembles of partitions via data resampling*, Proc. Intl. Conf. on Information Technology, ITCC 04, Las Vegas, 2004.
31. B. Minaei-Bidgoli, A. Topchy, W.F. Punch, *A comparison of resampling methods for clustering ensembles*, Proc. Intl. Conf. Machine Learning Methods Technology and Application, MLMTA 04, Las Vegas, 2004.
32. A. Mirzaei, M. Rahmati, and M. Ahmadi, *A new method for hierarchical clustering combination*, Intell. Data Anal. 12 (2008), 549–571.
33. M. Mojarad et al., *Consensus function based on clusters clustering and iterative fusion of base clusters*, Int. J. Uncertainty Fuzziness Knowl. Based Syst. 27(1) (2019), 97–120.
34. M. Moradi et al., *CMCABC: Clustering and memory-based chaotic artificial bee colony dynamic optimization algorithm*, Int. J. Inf. Technol. Decis. Mak. 17(04) (2018), 1007–1046.
35. A. Nazari et al., *A comprehensive study of clustering ensemble weighting based on cluster quality and diversity*, Pattern. Anal. Applic. 22(1) (2019), 133–145.
36. S. Nejatian, H. Parvin, and E. Faraji, *Using sub-sampling and ensemble clustering techniques to improve performance of imbalanced classification*, Neurocomputing 276 (2018), 55–66.
37. A. Y. Ng, M. I. Jordan, and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, in *Advances in Neural Information Processing Systems (NIPS)*, 2002, 849–856.
38. M. N. Omidvar et al., *A new natural-inspired continuous optimization approach*, J. Intell. Fuzzy Syst. 35(3) (2018), 3267–3283.
39. H. Parvin, A. Beigi, and N. Mozayani, *A clustering ensemble learning method based on the ant colony clustering algorithm*, Int. J. Appl. Comput. Math. 11(2) (2012), 286–302.
40. H. Parvin and B. Minaei-Bidgoli, *A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm*, Pattern Anal. Appl. 18 (2015), 87–112.
41. H. Parvin et al., *Data weighing mechanisms for clustering ensembles*, Comput. Electr. Eng. 39 (2013), 1433–1450. https://doi.org/10.1016/j.compeleceng.2013.02.004.
42. H. Parvin et al., *A new classifier ensemble methodology based on subspace learning*, J. Exp. Theoret. Artif. Intell. 25 (2012), 227–250. https://doi.org/10.1080/0952813X.2012.715683.
43. H. Parvin, S. Nejatian, and M. Mohamadpour, *Explicit memory based ABC with a clustering strategy for updating and retrieval of memory in dynamic environments*, Appl. Intell. 48(11) (2018), 4317–4337.
44. J. M. Peña, J. A. Lozano, and P. Larrañaga, *An empirical comparison of four initialization methods for the* K*-means algorithm*, Pattern Recogn. Lett. 20 (1999), 1027–1040.
45. V. Singh et al., *Ensemble clustering using semidefinite programming with applications*, Mach. Learn. 79 (2010), 177–200.
46. A. Strehl and J. Ghosh, *Cluster ensembles: A knowledge reuse framework for combining multiple partitions*, J. Mach. Learn. Res. 3 (2003), 583–617.
47. A. Topchy, A. K. Jain, and W. F. Punch, *Combining multiple weak clusterings*, Proc. 3d IEEE Intl. Conf. on Data Mining, 2003, pp. 331–338.
48. A. Topchy, A. K. Jain, and W. F. Punch, *A mixture model for clustering ensembles*, Proc. SIAM Intl. Conf. on Data Mining, SDM 04, 2004, pp. 379–390.
49. A. Topchy, B. Minaei-Bidgoli, A.K. Jain, and W.F. Punch, *Adaptive clustering ensembles*, Proc. Intl. Conf on Pattern Recognition, ICPR, Cambridge, UK, 2004.
50. T. Wang, *CA-Tree: A hierarchical structure for efficient and scalable coassociation-based cluster ensembles*, IEEE Trans. Syst. Man Cybern. B Cybern. 41 (2011), 686–698.
51. H. Wang, H. Shan, and A. Banerjee, *Bayesian cluster ensembles*, Proceedings of the Ninth SIAM International Conference on Data Mining, 2009, pp. 211–222.
52. M. Yasrebi et al., *Optimisation inspiring from behaviour of raining in nature: droplet optimisation algorithm*, Int. J. Bio-Inspired Comput. 12(3) (2018), 152–163.
53. Z. Yu et al., *Hybrid clustering solution selection strategy*, Pattern Recogn. 47 (2014), 3362–3375.
54. Z. Yu et al., *Adaptive noise immune cluster ensemble using affinity propagation*, IEEE Trans. Knowl. Data Eng. 27 (2015), 3176–3189.
55. X. Zheng et al., *Instance-wise weighted nonnegative matrix factorization for aggregating partitions with locally reliable clusters*, Proc. IJCAI (2015), 4091–4097. https://dblp.org/rec/conf/ijcai/ZhengZGM15.html
56. C. Zhong et al., *A clustering ensemble: Two-level-refined co-association matrix with path-based transformation*, Pattern Recogn. 48 (2015), 2699–2709.
57. Z. H. Zhou, *Ensemble methods: Foundations and algorithms*, Chapman & Hall/CRC, Boca Raton, FL, 2012 ISBN: 978-1-439-830031.
58. Z. H. Zhou and W. Tang, *Clusterer ensemble*, Knowl.-Based Syst. 19(1) (2006), 77–83.


**How to cite this article:** Najafi F, Parvin H, Mirzaie K, Nejatian S, Rezaie V. Dependability-based cluster weighting in clustering ensemble. *Stat Anal Data Min: The ASA Data Sci Journal*. 2020;13:151–164. https://doi.org/10.1002/sam.11451