Engineered Science
DOI: https://dx.doi.org/10.30919/es1186

Tri-Level Robust Clustering Ensemble (TRCE) Algorithm for Clustering Enhancement with Comparative Analysis

Jamil AlShaqsi,[1,*] Wenjia Wang[2] and Osama Drogham[3, 4]

Abstract

This paper introduces a Tri-Level Robust Clustering Ensemble (TRCE) algorithm that leverages the Three-Staged Clustering Algorithm (TSCA) as a fundamental component in its advancement. The purpose of TRCE is to enhance the clustering outcomes' quality by initially generating multiple clustering outcomes at various θ values. Subsequently, a voting mechanism is employed to consolidate the outcomes produced by each θ value. The fundamental aim of such a process is to ensure that the generated clustering results have maximized the ratio of agreements between the samples within each generated cluster. The TRCE has been verified on some benchmark datasets and then compared with several individual algorithms such as K-means, K-prototype, and squeezer algorithms. It has also been compared with ensemble-based algorithms such as selecting initial seeds based on the co-association matrix (SICM), selecting initial seeds based on previous results (SIPR), Average Normalized Mutual Information (K-ANMI), Clustering Categorical Dataset (CDC) and Clustering Categorical Data by Cluster Ensemble (ccdByEnsemble). Some supervised clustering algorithms were also involved in the comparison such as ICKDC, SKDEKMean, and ISSKDEKMeans. The experimental results showed the strengths of the TRCE in terms of the clustering quality over the compared clustering algorithms. It was always ranked highly and managed to handle numerical, categorical, and mixed-type datasets.

Keywords: Clustering; Clustering ensemble; Similarity measure; Cluster quality; Intra-Similarity; Inter-Similarity; Number of clusters.
Received: 15 May 2023; Revised: 18 January 2024; Accepted: 20 January 2024.
Article type: Research article.

1. Introduction

Clustering analysis is a data mining technique that has been studied for several years for data exploratory.[1-3] Its fundamental task is to segment a given dataset into several subgroups based on some similarities.[4-6] It is considered one of the challenging approaches in the area of data mining as in most cases it does not have internal criteria to measure the accuracy of the generated results. A clustering ensemble is a method for combining clustering algorithms with a decision function in the best way possible to get more reliable and high-quality results.[7,8] There are two main approaches to building a clustering ensemble. The first method involves using a set of individual clustering algorithms, *H*, to produce different clustering results and then aggregate the generated results. The second method involves using a single algorithm and running it several times with various initial seeds or dissimilar parameters and then aggregating the clustering results. Each approach has its method to aggregate the generated results.

As for the first approach, primarily each algorithm is run against the given dataset to generate the initial results. Then a voting mechanism will play the role of aggregating the results of each component. For such an approach, generally, the Similarity-Graph algorithm[9] is used to combine the initial results. Indeed, other potential difficulties need attention while using this strategy. The first concern arises from the

[1] *Information Systems Department, Sultanate Qaboos University, Muscat, P.O. Box: 20, P.C: 123, Sultanate of Oman.*
[2] *School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK.*
[3] *Prince Abdullah bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University, Al-Salt, 19117, Jordan.*
[4] *School of Information Technology, Skyline University College, University City of Sharjah, Sharjah, P.O. Box 1797, United Arab Emirates.*
**Email:* alshaqsi@squ.edu.om (J. AlShaqsi)

unlikelihood of achieving a comprehensive partition due to the potential for disparate outcomes resulting from distinct clustering criteria used by each method, denoted as H. Hence, it is a challenging task to choose which algorithm's outcomes should be considered. Another significant concern is the absence of rules about the optimal number of algorithms, denoted as H, that should be included in the formation of the clustering ensemble. The second way of aggregating the results is using diverse initializations to execute an algorithm several times on a dataset. Subsequently, a voting process is used to amalgamate the produced outcomes into a singular, consolidated clustering result. The convergence clustering ensemble, which is based on the *K*-means algorithm, implements this approach.[10] Whereas, Random-*K* and Random-K+[11] implement the second approach of the clustering ensemble. The basic idea of the Random-*K* algorithm is to execute the same algorithm several times against a dataset. To increase ensemble diversity, various values for the number of clusters *K* are used. Normally, the value is within the range of 2 to *K*+10.[12] The concept of the Random-*K*+ algorithms is more or less similar to that of the Random-*K*; expect that it has a larger interval for the number of clusters, *K*+30. Concerning the drawbacks of this method: (1) there is no confirmed standard to identify the value of *H*, (2) determining the value of *K* is considered a possible issue in this approach, (3) still there is no justification for selecting the range for the *K* value such as *K*+10 and *K*+40. Thus, this method is not practical and probably less effective in real applications.

Several clustering techniques have been developed and employed in several applications.[5,13,14] However, clustering still be classified as one of the most challenging techniques as; generally, each algorithm has its constraints in some aspects. Thus, none of them can sufficiently address the various problems of clustering and generate reliable results. Each clustering method has its distinct resilience in addressing certain parts of clustering issues, which might provide challenges for other algorithms. This strength potentially compensates for the flaws of the other algorithms. Hence, it is crucial to create a clustering ensemble by merging separate algorithms to collaborate effectively to get optimum clustering outcomes.

It has been demonstrated that, in terms of clustering quality, clustering ensembles are more accurate than the individual clustering algorithms in some studies.[11,12,15-20] Nevertheless, when implementing such a technique, three important issues need to be considered to ensure an optimal ensemble result: (1) diversity among the individual components.[11,18,19] (2) the accuracy of each clustering algorithm, and (3) the aggregating mechanism. Diversity can be obtained by many heuristics.[21-24] Utilizing a random selection of target clusters for each member of the ensemble is regarded as a very effective heuristic in ensemble design.[25] There are several difficulties linked to the design of the ensemble. For instance, finding an effective approach to combine the partition results. Accomplishing this job is challenging due to the absence of pointers in data samples. Furthermore, there is a lack of explicit instructions about the organizational framework and the optimal quantity of components required for constructing the ensemble. This difficulty has a substantial impact on the overall performance of the clustering ensemble's produced output. This study introduces a framework called the TRCE, which builds upon the qualities of the TSCA as its main clustering generator. First, the TSCA will estimate the most probable number of clusters automatically for a given dataset. Then the clustering results of *H* sets will be generated. Finally, to optimize the clustering results, the voting mechanism will be employed to aggregate the results of each set.

The main contributions of this proposed algorithm are:

- Designing an ensemble algorithm (TRCE) based on the Three-Staged clustering algorithm (TSCA) to improve the quality of the clustering results.
- Aggregation of the generated clusters within a certain threshold range.
- Defining a novel criterion for calculating the *Inter-Cluster Similarity* (*InterS*) which represents the quality of cluster separations from each other.
- Development of *Intra-Cluster Similarity* (*AIntraS*) robustness measure to calculate the strength of the generated clusters.
- Estimating the appropriate number of clusters automatically.
- Sorting out the problem of cluster indexing; hence, clusters always have the same reference number within an interval.
- Producing dependable clustering outcomes. Therefore, running the procedure several times with consistent threshold values will provide the same clustering results, irrespective of the sample order.

The rest of this paper is organized as follows. The related work is highlighted and reviewed in Section 2. The details of the proposed TRCE are presented in Section 3. Section 4 presents the complexity of the TRCE. Sections 5 and 6 spot the light on the datasets, and the experimental results and evaluation, respectively. Section 7 presents an analysis and summary of conducted experiments. Finally, yet importantly, the paper will present the conclusion in Section 8 highlighting the fundamental issues raised.

2. Related work

The Three-Staged Clustering Algorithm (TSCA) has three primary phases. The primary objective of the first stage is to create the first clusters of individuals who are not friends with each other. The second stage is executed to optimize the accuracy by refining the generated clusters. In the last stage, the overall accuracy is enhanced by improving the *BASES* that were selected initially.

- **First stage**

In building up the initial clusters, first, a threshold value *(θ)* and a *BASE, B,* need to be chosen. A *BASE* is a data sample that is selected based on some calculations to represent a particular cluster. Once the *BASE* is found, the similarity between the *BASE* and each data sample is calculated. Equation 1, which is presented in the next section, is used to calculate the similarity. Samples that have a similarity value at least equal to the proposed threshold value will be allocated to the *BASE's* Cluster. Samples that are not assigned to any of the initial clusters will be used to form the next cluster with a new *BASE*. These processes will be repeated until there are no remaining data samples.

- **Second stage**

The process of refining the generated clusters commences by computing the similarity between the *BASE* of the second generated cluster and all the data samples that were allocated to the first cluster. If any data sample scored a better similarity value compared to its current cluster, such a sample has to be reallocated to its best cluster.

- **Third stage**

To verify whether the clustering results can be further improved or not the third stage commences by refining the initial *BASEs* and then processing the second stage again. This process will be repeated until no sample movements occur between the clusters. Equation 1, which is presented in the next section, is used to calculate the similarity between each cluster's initial *BASEs* and the constructed potential *BASEs*.

Zhou, *et al*.[26] developed an algorithm based on a clustering method called ICKDC. It is a supervised clustering algorithm and was tested on synthetic and benchmark datasets. To evaluate the accuracy of the clustering result, different evaluation metrics were used such as the Fowlkes Mallows index (FMI), Adjusted Rand Index (ARI), Adjusted Mutual Information (NMI), and the accuracy.[26] The accuracy r can be measured by $r = \frac{1}{n}\sum_{i=1}^{k} a_i$[27-33] where n is the number of data samples in the given dataset, and a_i represents the number of largest data samples with a similar label in cluster i, The clustering results confirmed that the proposed algorithm is not the absolute winner. Out of six cases, the proposed algorithm managed to get the best accuracy in two cases only.

The author in Ref. [34] proposed the SMKNN algorithm which is a clustering algorithm based on the K-nearest neighbor (KNN). The SMKNN algorithm follows the principle of split and merge and it consists of three stages. In the first stage, the KNN graph is constructed. The second stage commences by eliminating the pivot samples from the KNN graph. Eventually, the output of such a process is the construction of the subgraphs. The creation of the clusters is performed in the third stage. The SMKNN was tested on some synthetic and benchmark datasets, and then compared with several clustering algorithms mainly graph-based clustering algorithms including the standard *K*-means. The authors claimed that the SMKNN outperformed the compared algorithms. However, this algorithm is limited to the numerical datasets with arbitrarily shaped clusters. It has not been tested on gene expression, categorical and mixed type datasets.

The authors in Ref. [35] proposed a modified *K*-means algorithm (mdK-means) that is based on a new similarity measure. The proposed algorithm can detect and remove the outlier by employing the Tukey rules. To improve the efficiency and minimize the outlier's influence on the centroid selection, the outlier removal takes place before the actual clustering process. The mdK-means was tested on 9 multivariate datasets obtained from Ref. [36] and then compared with 8 different clustering algorithms. The authors claimed that the mdK-means outperformed the compared algorithms in terms of clustering accuracy. However, the mdK-means cannot handle categorical or mixed types datasets; thus, the experiments were limited to the numerical dataset. Besides, the selected datasets do not have many outliers; thus, the proposed feature of the outlier detection needs to be verified on their complex datasets with different level of noise.

Bortoloti, *et al*.[37] proposed an incremental semi-supervised *K*-means algorithm (ISSKDEKMeans). The ISSKDEKMeans has two main stages: offline and online stage. Concerning the offline stage, it employed the supervised *K*-means to construct the initial clusters. To enhance the clustering quality, the second stage commences, in which the supervised clustering algorithm (SKDEKMeans) is employed, to learn from the new data sample and update the initial clusters. To evaluate the accuracy of the ISSKDEKMeans algorithm, 16 benchmark datasets were used from different data repositories including UCI, SSL-Book, and KEEL benchmarks. To conduct the experiments, the dataset was divided into three portions: 20% labelled examples, 60% unlabelled examples, and 20% test examples. The experimental results indicated that the developed algorithm performed better than the compared

algorithm in 50% of the cases. This confirmed that the proposed algorithm has some limitations; thus, further investigation and development in clustering are required.

The *K*-means algorithm is an unsupervised learning algorithm.[38] It is computationally efficient with a run time complexity of O($K * n$), where K is the number of clusters and n represents the number of samples. It is constrains-less as it requires no constraints to be set in advance[39] except the value of K. However, it is very sensitive to the random seed initialization as poor clustering results might be obtained if inappropriate seeds are selected. Besides, a random selection of the value of K is considered one of its main weaknesses. This is because a small value of K might result in developing clusters with irrelevant samples.[12] In contrast, a large value of K may produce over fragmented clusters.[12] For the prediction of the best possible initial seeds, two methods were developed.[10] These methods may address the limitations associated with the arbitrary selection of the seed in the K-means algorithm. The initial seed selection strategy, known as SIPR, is based on the concept of choosing seeds based on previous outcomes. The steps of this technique include first using the K-means algorithm with random seeds to generate K clusters.[10]

The algorithm will be executed for several iterations given the fact that each developed cluster should propose seeds for the next iteration. Accordingly, clusters in iteration i propose the seeds for the clusters in iteration $i + 1$. The SICM is the abbreviation for "selecting initial seeds based on the co-association matrix".[9] It is the second method that was developed by Azimi J., *et al*.[10] It is based on the concept that if the co-association value between the samples is less than the threshold, such samples will be used as the initial seeds. The co-association matrix is updated after each execution of the *K*-means algorithm.

Another co-association matrix to merge numerous clustering results was developed by the authors in [12] In this matrix, the result of the co-association determines the degree of robustness of the association between data samples. Accordingly, clusters are developed by combining data samples that have a co-association value more than the specified threshold. Like the case in [9], the voting mechanism is employed to determine the clustering result. As for future work, the authors aim to enhance the co-association values by utilizing the benefits of the clustering algorithm(single-link) rather than relying on a fixed predefined threshold.[12] The authors in Ref. [40] utilize the feature of the co-association and take it one level up to form a clustering ensemble. However, in the ensemble, only samples with a high value of co-association are considered to form the clusters. Therefore, the final results will contain fewer samples than the processed dataset as some samples, with co-association less than the set threshold, will be excluded.

The authors in Ref. [21] employed one of the hierarchical clustering algorithms (single linkage) to aggregate the results of several executions of the *K*-means algorithm. To determine the optimal clustering result, a genetic algorithm was proposed by the authors in Ref. [41] to aggregate the generated results. To define the association value between the labels of each clustering and those of an optimal meta-clustering, the authors in Ref. [42] used linear programming. The authors in Ref. [43] developed an aggregation mechanism for clustering. Two stages of experiments were conducted by the authors in [18]. First, the authors ran several experiments on soft clustering and then applied the clustering aggregation to obtain random projections.

Strehl and Ghosh considered three different consensus methods for the clustering results.[44] The first method applies to the concepts of the evidence accumulation approach, which was proposed in [20,21]. The other two methods are based on the idea of the hypergraph representation. The Cluster-based Similarity Partitioning Algorithm (±) employs an algorithm called METIS[45] to bring a graph from a co-association matrix and clusters.

To cluster the categorical dataset, an ensemble method was proposed on the concept of executing parallel algorithms to cluster a given dataset.[46] Subsequently, the technique that produces the most Average Normalized Mutual Information (ANMI) was chosen as the most ideal outcome. Three hypergraph-model-based methods, including the HyperGraph Partitioning approach (HGPA), CSPA, and Meta-Clustering Algorithm (MCLA), were modified for the suggested approach.

The authors in [47] proposed a technique to cluster mixed datasets: numerical and categorical. The technique is called Divide-and-Conquer and it commences by splitting the given dataset into two subclasses: numerical and categorical. Subsequently, a series of clustering algorithms are used for each subclass to provide preliminary outcomes. Ultimately, the findings acquired from the two subcategories are combined to create a category dataset. The technique was evaluated on benchmark datasets (Credit Approval and Cleve) that included a combination of numerical and categorical variables. The "Clustering Categorical Data By Cluster Ensemble (ccdByEnsemble)", aims to combine several clustering ensembles and then utilize the benefits of the CSPA to determine the final results.[46] The ultimate aim is to generate high-quality clusters by running several clustering algorithms in parallel including HGPA,[48] CSPA[25] and MCLA on the

same dataset. Once the results are generated, a comparison on the value of the ANMI is performed and the one that generates the highest ANMI[46] is selected as a final partition. To evaluate the quality of the ccdByEnsemble, several experiments were conducted on benchmark datasets (Zoo, Cancer, Votes and Mushroom). The obtained results were then compared with other algorithms for example Squeezer and GAClust.[49] The ccdByEnsemble won in two cases (Cancer and Votes datasets) and lost in the other two cases (Zoo and Mushroom sets). Generally, it performed somehow the same level of accuracy as the compared algorithms.

The authors in [50] proposed the *K*-ANMI algorithm. According to the authors, this algorithm is appropriate for clustering categorical datasets. The performance of the *K*-ANMI algorithm was evaluated by using the ANMI measure as the criterion. The authors tested the *K*-ANMI algorithm on 3 benchmark datasets (Cancer, Mushroom and Votes). The experimental results were then compared with other algorithms. The authors claimed that *K*-ANMI outperformed the 4 compared algorithms. However, *K*-ANMI also has some limitations such as it needs the value of *K* to be provided prior to the actual clustering process the be used as the basis of finding optimal clustering results.

3. Proposed ensemble method for clustering

The proposed TRCE commences by generating the initial sets of clustering results. Then aggregation method is used to highlight the final clustering results at each threshold value (θ). The θ value that generates high average intra-cluster similarity ($AIntraS_\theta$) will be qualified for voting given the fact that it is higher than or equal to the average intra-cluster similarity of the interval ($AIntraS_I$). This will maximize the intra-cluster similarity (*IntraS*) which is the similarity between the samples within a cluster. At the same time, it will maximize the dissimilarity between the generated cluster, and the inter-cluster similarity (*InterS*). Finally, a voting mechanism is employed in each qualified interval.

The main steps of the TRCE algorithm are:

1. Execute the TRCE

1.1 Find the candidate intervals, *I,* for the given dataset by running the TRCE from θ value ranges from 1% to 100%.

1.2 Once the TRCE starts generating small interval lengths continuously, L < 2, terminate the process.

1.3 Highlight the candidate intervals and the one that generates appropriate intra-cluster similarity (*IntraS*) and inter-cluster similarity (*InterS*) will be selected.

2. Re-execute the TRCE for each θ in the selected intervals *I*

2.1 Nominate a *BASE* to build the first cluster:

a. For the numerical features, find the centroid by calculating the average

b. Find the most frequent category in each categorical feature: calculate the frequency of each category and then highlight the highest value, mode.

2.2 Based on the obtained centroid and modes, build a temp sample.

2.3 Find the similarity between the temporary sample and the real samples in the potential dataset by using Equation 1 below.

$$Sim\,(x_i, B_k) = \frac{1}{N}\left[\left(\begin{cases}\sum_{j=1,for\ x_{ij}\in R}^{N}\left(1-\frac{|x_{ij}-B_{kj}|}{\max\{|x_{ij}|,|B_{kj}|\}}\right)\\ 0\ if\ x_{ij}=B_{kj}=0;\end{cases}\right)+\left(\sum_{j=1,for\ x_{ij}\in Cat}^{N}\begin{cases}1\ if x_{ij}=B_{kj}\\ 0\ if x_{ij}\neq B_{kj}\end{cases}\right)\right] \quad (1)$$

where Sim represents the similarity, x represents the sample; B is the BASE, N is the number of features, j is feature index, i is sample index; k is the index for BASES and clusters. Cat and R represent the categorical and the numerical features; respectively.

2.4 The sample that will have the highest value with the temporary constructed sample will be considered as the BASE.

2.5 Run the TRCE by calculating the similarity between the nominated BASE and the samples in the dataset by using Equation 1.

2.6 Construct the first cluster for the first nominated BASE by assigning the samples that have similarity value at least equal to the set threshold θ.

2.7 The remaining samples, if any, will be used to nominate a new BASE for the next cluster.

2.8 Steps 1 to 7 will be repeated until no data samples left.

2.9. Repeat steps 1 to 8 for the remaining threshold values θ within the selected interval.

3. Clustering ensemble

1. For each threshold value, θ, within the selected interval, the TRCE has generated *K* clusters.

2. Compute the *IntraS* of each K_j cluster by using equation (2). In this equation ***S*** represents the number of samples in each K_j.

$$IntraS_k = \frac{1}{S_k}\left(\sum_{i=1\&k=1,\in R}^{k} Sim(x_{ik}, B_k)\right) \quad (2)$$

3. Calculate the $AIntraS_\theta$ for the first threshold value by dividing the obtained $IntraS_k$ by number of clusters *K* as given in equation (3).

$$AIntraS_\theta = \frac{1}{k}\left[\sum_{k=1}^{k}(IntraS_k)\right] \quad (3)$$

4. Equation (4) presents the calculation of the $AIntraS_I$ for interval *I*.

$$AIntraS_I = \frac{1}{L}\left[\sum_{\theta=L_{Start}}^{L_{end}}(AIntraS_\theta)\right] \quad (4)$$

5. Concerning the voting mechanism, if the obtained $AIntraS_\theta$ is at least equal to the $AIntraS_I$, ($AIntra_\theta \geq AIntraI$) then the

clustering results of such a threshold value will be qualified for the voting.
6. Steps 1 to 5 are repeated for the remaining θ values.
7. Finally, the voting mechanism will be employed to generate the final partition which will ensure the highest $AIntraS_I$ and valid *InterS*

To illustrate how a *BASE* is nominated, consider the example in Table 1 which presents an example of the categorical feature.
1. Calculate the frequency of the category in each feature as shown in Table 2.
2. Construct a temporary artificial sample by considering the most frequent category in each feature as show in Table 3.
3. Find the similarity between the artificial sample and all real samples in the dataset by using Equation (1) above.
4. The nominated *BASE* will be the sample that will have the highest similarity value with the temporary artificial sample.

The systematic approach of the TRCE for choosing a *BASE* for each cluster enables it to overcome the problem of cluster indexing which is the concern in most other clustering algorithms. This is because the method of choosing a *BASE* confirms that clusters always have the same reference number within an interval. Additionally, it guarantees that the method will provide dependable clustering results. If the TRCE is repeatedly conducted with the same threshold value, θ, it will consistently provide the same clustering outcomes.

Table 1. Balloon dataset.

Gender	Address	Qualification	Married	Buy
Male	Salalah	PhD	No	No
Male	Muscat	BSc Degree	Yes	Yes
Male	Muscat	BSc Degree	Yes	No
Male	Muscat	BSc Degree	No	Yes
Male	Muscat	PhD	Yes	Yes
Male	Salalah	BSc Degree	No	Yes
Male	Salalah	BSc Degree	Yes	Yes
Male	Salalah	PhD	Yes	Yes
Female	Muscat	BSc Degree	No	No
Female	Muscat	BSc Degree	Yes	No

Table 2. Frequency of each feature.

Feature	Frequency	
Gender	Male = 8	Female = 2
Address	Salalah = 4	Muscat= 6
Qualification	PhD = 3	BSc Degree = 7
Married	Yes = 6	No = 4
Buy	Yes =6	No = 4

Table 3. Constructing of a temporary artificial sample.

Gender	Address	Qualification	Married	Buy
Male	Muscat	BSc Degree	Yes	Yes

Table 4 presents a working example about the voting mechanism of the TRCE algorithm. The given dataset has tens samples $DS = \{D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9, D_{10}\}$, where *D* represents the samples. The obtained clustering results (CR) are as follow: $CR_1=[\{D_2, D_6\}, \{D_1, D_3, D_4, D_{10}\}, \{D_5, D_8\}, \{D_7, D_9\}]$, $CR_2=[\{D_3, D_6\}, \{D_1, D_2, D_4, D_{10}\}, \{D_5, D_8, D_9\}, \{D_7\}]$, $CR_3=[\{D_1, D_2, D_3, D_4, D_5, D_8, D_{10}\}, \{D_7, D_9\}\}$, $CR_4 = [\{D_1, D_2, D_6, D_{10}\}, \{D_3, D_8\}, \{D_4, D_5, D_9\}, \{D_7\}]$, and $CR_5 = [\{D_1, D_2, D_6, D_{10}\}, \{D_3, D_8\}, \{D_4, D_5\}, \{D_7, D_9\}]$. The symbol (✗) indicates that practical *CR* is not qualified for the voting; whereas the tick symbol (✓) means the CR will participate in the voting. The vote results are shown in the voting column as a Cluster Ki (frequency). The frequency measures the number of times a sample was assigned to Ki, taking into account just the votes for the qualifying outcomes. As an example, the data sample, D1, was allocated three times to cluster 2 and two times to cluster 1. However, the clustering results of CR2 and CR3 are not qualified for the vote, therefore they are given as 2(1) and 1(2) correspondingly, instead of 2(3) and 1(2). Therefore, none of them participated in the voting process. The final partition (FP) is the best clustering outcome obtained via the process of voting (max. voting) for each sample.

Table 4. The Concept of the TRCE.

DS	✓	✗	✗	✓	✓	Voting	*FP*
	CR_1	CR_2	CR_3	CR_4	CR_5		
D_1	2	2	2	1	1	2(1), 1(2)	1
D_2	1	2	2	1	1	1(3)	1
D_3	2	1	2	2	2	2(3)	2
D_4	2	2	2	3	3	2(1), 3 (2)	3
D_5	3	3	2	3	3	3(3)	3
D_6	1	1	3	1	1	1(3)	1
D_7	4	4	3	4	4	4(3)	4
D_8	3	3	2	2	2	3(1), 2(2)	2
D_9	4	3	3	3	4	4(2), 3(1)	4
D_{10}	2	2	2	1	1	2(1), 1(2)	1

One of the good feature of the TSCA its capability in estimating the optimal value of clusters, *K*. This is achieved by highlighting the candidate intervals *I*. The value of L represents how many times the TSCA generates a constant value of *K* continuously and is calculated as $L = (L_{End} - L_{Start}) + 1$, where L_{Start} and L_{End} represent that beginning and the end of the L, respectively. The experiments have proven that there is a positive relationship between the $AIntraS_\theta$ and the clustering accuracy. High $AIntraS_\theta$ always associated with high accuracy. More details about estimating *K* value are presented in [51].

The TRCE algorithm aims to converge the local maximum by addressing the following points:
1. After generating the initial clusters, the samples in each generated cluster will nominate a new *BASE*, called *Proposed BASE*, following the nomination procedures above.
2. Refine initial *BASEs* by calculating the similarity between the *BASE* in each cluster and the new *Proposed* corresponding *BASE*.
3. If the similarity results in a value less than 1, the initial and the proposed *BASEs* are not identical, replace the current

BASEs with the *Proposed BASEs* and rerun the TRCE.
4. Terminate the clustering process once the initial *BASEs* and the proposed *BASEs* are identical, similarity is approaching 1.

4. Complexity of the proposed algorithm

In the TRCE algorithm, most of the time is spent in calculating the similarity between the samples and the *BASEs* of the generated clusters. In this step, each sample gets added to its most similar *BASE* once. As a *BASE* is a real sample, one such operation costs $O(K*(n - K))$. In refining the *BASEs,* the TRCE algorithm iterates through all the K clusters *BASEs* and tries to replace each of them with one of the remaining $(n - K)$ samples. If no replacement is done, the algorithm will be terminated. On the other hand, if a replacement is occurred, the next iteration will take place. Since the TRCE algorithm attempts to replace each of K *BASEs*, initial *BASE* B_K, with one of the $(n - K)$ samples, potential *BASE* Q_k, and each of these attempts results in $(n - K)$ operation, the complexity for each iteration is $O(I_\theta * K* (n - K)^2)$, *where* I_θ is the number of iteration in each interval and it will be represented as:

$$\sum_{i=1}^{L} O(I_i * \mathrm{K}*(\mathrm{n} - \mathrm{K})^2)$$

Although such an operation is very expensive for large datasets, it is less sensitive to noise and outlier as the algorithm uses *BASEs* instead of artificial centroids.

5. Datasets

To validate the effectiveness of the TRCE, it was compared with some of the existing clustering algorithms. To ensure a fair and valid comparison with existing algorithms, the benchmark datasets were limited to the data that were involved in the experiments of the previous researchers. The demographic details of the used benchmark datasets are presented in Table 5. These datasets were obtained from the UCI data repository.[36] The datasets were selected because of their different types (numerical, categorical and mixed data), different sizes, the diverse number of clusters K and the different dimensions. No data cleansing was conducted; however, in the Credit Approval dataset, the missing values were removed whereas the missing values in the Cleve dataset were replaced with the value of ZERO "0". The class labels of each dataset were not included in calculating the similarity.

6. Excremental results and evaluation

To prove the robustness of the proposed TRCE algorithm, it has been tested on the above benchmark datasets and then compared with a number of the existing algorithms. In the experiments of the compared algorithms, it is noticeable that in some cases, different K values were used by the original authors for numbers of clusters. The values of K range from 2 to 9 as some of the compared algorithms do not have the ability to detect the ideal number of clusters. Therefore, to make fair comparisons with the compared algorithms, the value of K that is estimated by the TRCE is confirmed for the experiments. Tables 6 to 11 present the comprehensive comparisons of the clustering accuracy between the TRCE and the compared algorithms.

Table 5. Details of the benchmark datasets.

No.	Datasets	Classes	Samples	Features	
				N	C
1	Mushroom	2	8124	0	22
2	Votes	2	435	0	16
3	IRIS	3	150	4	0
4	Wine	3	178	13	0
5	Soybean	4	47	35	0
6	Credit Approval	2	690	6	9
7	Cleve	2	303	6	8
8	Cancer	2	699	0	9
9	Zoo	7	101	1	15
10	Half-rings	2	400	2	0
11	2-spirals	2	200	2	0
12	Heart	2	267	23	0
13	Glass	6	214	9	0
14	Balance	3	625	4	0
15	Vertebral	2	310	6	0
16	Ecoli	5	327	5	0
17	Blood	2	748	4	0
18	Seed	3	210	7	0

Table 6 compared the accuracy of the TRCE with the experimental results given by the authors in [10]. For Iris dataset, TRCE algorithm achieved the best accuracy. The SICM scored an accuracy of 90.45% which is ranked as the second best. In Wine dataset, the TRCE algorithm performed approximately 20% better than SIPR, SICM and K-means. In clustering the Soybean dataset, the algorithms used by the authors in [10] did not succeed to obtain an accuracy higher than 77%. In contrast, the TRCE algorithms scored an accuracy of 100%.

Table 6. Accuracy comparison on iris dataset, wine and soybean.

Dataset	*K*	SIPR	SICM	*K*-means	TRCE
Iris	3	88.23	90.45	77.78	95.3
Wine	3	72.47	75.28	70.78	93.3
Soybean	4	73.06	76.59	68.08	100

The GAClust, squeezer, ccdByEnsemble and K-ANMI algorithms were selected to cluster the Cancer, Votes, Zoo and Mushroom datasets.[46,50] According to the experimental results in Table 7, the TRCE achieved the best accuracy for the numerical datasets: Cancer and Votes. For the Zoo dataset, which is a mixed data type with an unbalanced class distribution, the TRCE algorithm scored the first position at an accuracy of 93.1%. None of the other compared algorithms manage to reach an accuracy of, at least, 90%. The K-ANMI is not tested on this dataset as it cannot handle mixed-type datasets. For the Mushroom dataset, the compared algorithms

obtained accuracies less than 70%. Like the above case, the TRCE algorithm is the absolute winner as it obtained an accuracy of 89%. This accuracy is almost 30% better than GAClust, Squeezer, and *K*-ANMI, and 20% better than ccdByEnsemble.

Table 7. Accuracy comparison on cancer, votes, zoo and mushroom datasets.

Datasets	*K*	GAClust	Squeezer	ccdBy Ensemble	*K*-ANMI	TRCE
Cancer	3	≈80	≈90	≈93	96	97
Votes	2	80	62	86	87	87
Zoo	8	≈87	≈89	≈85	-	93
Mushroom	2	61	54	67	59	89

Table 8 presents the experimental results of Credit Approval and Cleve. Both of datasets are of mixed type dataset. Concerning the clustering results, it has been presented that the algCEBMC algorithm yielded the best clustering accuracy by its original authors.[47] Thus, this algorithm is selected for the comparison. In clustering the Credit Approval dataset, TRCE achieved the best clustering accuracy at an accuracy of 79%. For the Cleve dataset the algorithms performed more or less the same. TRCE generated the highest accuracy. The algCEBMC scored the second-best clustering results. So, in both cases the TRCE outperformed the algCEBMC clustering algorithm.

Table 8. Accuracy comparison on credit approval and cleve datasets.

Datasets	*K*	K-prototypes	algCEBMC	TRCE
Credit Approval	2	≈73	≈77	79
Cleve	2	≈78	≈84.2	85

The TRCE algorithm is also compared with five other algorithms to clustering the Half-rings, Iris, and 2-spirals datasets.[52] For Half-rings dataset, the TRCE algorithm identified 3 clusters; thus, this number is confirmed for the comparison. Concerning the clustering accuracy, as presented in Table 9, the TRCE algorithm achieved the best at an accuracy of 87.8%. In some scenarios when the number of ensemble components (H = 5) is low, the EM method achieved the second highest level of accuracy. The MCLA achieved the third highest performance in some instances when the value of H is equal to or greater than 10. The clustering results of the EM and QMI algorithms confirmed that high value of *H* does not guarantee high accuracy.

Table 10 lists the performance of the TRCE algorithm compared with the other five algorithms. As presented the TRCE algorithm beats the compared algorithms by more than 10%. The CSPA achieved the second-highest level of accuracy when H was equal to or greater than 15. However, it is not advisable to use large values of H in practical applications. The empirical findings about the 2-spirals dataset may be seen in Table 11.[52] The TRCE algorithm has determined that the expected value of k for this dataset is 7. At this particular value of K, although other algorithms used high quantities of H, the TRCE method had the highest clustering performance.

Table 9. Accuracy comparison on half-rings dataset.

H	*K*	EM	QMI	HGPA	CSPA	MCLA
5	2	74.6	74.6	50	74.5	74.6
5	3	76	63.2	51.2	73.8	74.9
10	2	73.3	66.8	50	71.4	76.3
10	3	66.5	60.3	74	75.1	75.8
30	2	73.1	59.4	50	73.8	74
30	3	70.7	64.1	72.5	73.8	73.8
50	2	72.8	67.7	50	70.5	78.9
50	3	71.2	64.7	75.2	75	75.4
-	3	TRCE: 87.8				

Table 10. Accuracy comparison on iris dataset.

H	*K*	EM	QMI	HGPA	CSPA	MCLA
5	3	89	85.3	58.6	88.8	89.1
10	3	89.2	89.2	61.8	88.7	89.1
15	3	89.1	88.1	57.2	90.2	88.9
20	3	89.1	85.5	60.9	90.2	89.1
30	3	89.1	87.2	56.6	92.1	88.7
40	3	89	87.6	58.1	92.3	88.9
50	3	89.1	86.2	57.3	92.1	88.8
-	3	TRCE : 95.3				

Table 11. Accuracy Comparison on 2-spirals dataset.

H	*K*	MCLA	HGPA	CSPA	QMI	EM
5	2	56.2	50	56.1	56.4	56.5
5	3	59.5	50.5	60.1	58.7	58.9
5	5	60	57	60	59	58.8
5	7	56.3	57.6	54.6	54.6	54.1
5	10	56.1	53.6	52.3	54.6	52.7
10	2	56.1	50	56	56.3	56.6
10	3	58.3	50.8	61	60	63.1
10	5	61.1	59.4	61.7	60.6	61.4
10	7	54.3	57	53.8	53.3	53.3
10	10	57.6	52.9	52.3	54.4	53.3
20	2	56.1	50	56.2	56.4	56.7
20	3	60	50.7	62.9	59.8	59.3
20	5	61.9	60	61.8	60.5	61.4
20	7	55.8	55.6	53.3	52.4	54.1
20	10	57.8	52.7	51.3	52.8	51.8
-	7	TRCE: 62.5				

TRCE is also compared with a recent work proposed by the authors in [26]. In the first paper, the authors used three algorithms which are ICKDC, SKDEKMean, and ISSKDEKMeans and six datasets (Cancer, Heart, Iris, Votes, Wine and Ionoshere). The Ionosphere dataset is not included in the experiments as the TRCE estimated the *K* value higher than that of the authors' experiments. ICKDC, SKDEKMean,

and ISSKDEKMeans algorithms are supervised clustering algorithms. This means that they have the advantage of training the model with few samples before the actual clustering takes place which is not the case in the TRCE. The experimental results of this comparison are presented in Table 12. As shown the TRCE achieved the best clustering results in the Cancer and Iris datasets. In clustering the Heart dataset, both the TRCE and ICKDC achieved the highest accuracy. The ISSKDEKMeans and SKDEKMean achieved the second and third best, respectively. For the Votes and Wine datasets, although the TRCE did not score the first place, it obtained competitive clustering results, given the fact it is an unsupervised clustering algorithm. In the Wine dataset, it achieved an accuracy of more than 90%.

Table 12. Accuracy comparison on cancer, heart, iris, votes, and wine datasets.

Datasets	*K*	ICKDC	SKDE-KMeans	ISSKDE-KMeans	TRCE
Cancer	2	94.02	94.08	93.51	96.7
Heart	2	79.4	73.68	74.62	79.4
Iris	3	92	94.33	94.33	95.3
Votes	2	88.2	92.27	93.85	84.6
Wine	3	93.82	96.11	97.78	93.3

Table 13 compared the accuracy of the TRCE with the experimental results obtained by the authors in [34]. The sign "–" indicates that the algorithm is unable to generate the required number of clusters. Although the authors conducted experiments on synthetic and real-world datasets, due to the unavailability of the synthetic datasets, the comparison will be limited to the real-world datasets. It has been presented by the authors that their proposed algorithms, SMKNN, yield the best clustering accuracy in most cases. Thus, this algorithm will be selected for the comparison with the TRCE. In clustering the Sonar and Soybean datasets, SMKNN and REC achieved the same clustering accuracy 59.6% and 100%, respectively. In clustering the Glass and Page-Blocks datasets, the accuracy of TRCE was slightly lower than the SMKNN algorithm. Its accuracy was lower by 3.3% and 1.7%, in the Glass dataset and Page-Blocks datasets, respectively. The TRCE outperformed the SMKNN in Semeion and SCADI datasets by more than 15%.

The authors in [35] conducted comprehensive experiments in *K*-means alike algorithms on 10 real-world datasets. As shown in Table 14 none of the compared algorithms is the absolute winner. The TRCE won in three cases and drew in one case. However, when it was beaten, its accuracy was lower by a tiny fraction. Whereas, when it scored the first position, it was higher by 15.6% and 7.6% higher in Balance and Icoli datasets, respectively. Overall, TRCE is considered as the best algorithm among the compared algorithms as it scored the best average clustering accuracy.

7. Summary of the experimental results

To ensure a fair and valid comparison between the compared algorithms and the REC, the results are displayed in seven different tables: 15 to 21. This is because the original authors did not test the compared algorithms on all datasets listed in

Table 13. Accuracy comparison with the SMKNN algorithm.

Algorithms	Datasets							
	Sonar	Soybean	Glass	Ionosphere	Page-Blocks	Semeion	SCADI	HillValley
K-means	54.6%	87.0%	42.2	71.1	48.1	53.8	63.0	50.5
SL	52.9	83.0	36.9	64.4	89.9	10.5	65.7	50.5
DPC	52.9	79.0	40.2	68.1	73.4	36.9	45.7	50.5
GDD	1.0	9.0	35.1	62.4	73.6	–	10.0	40.4
SPC	52.9	90.0	40.7	64.4	89.8	11.0	65.7	50.4
SAM	51.4	98.0	36.4	58.1	44.4	67.0	68.6	51.0
cciMST	53.4	89.0	37.4	64.4	35.8	59.5	57.1	50.5
Chameleon	56.3	91.0	43.5	68.1	52.9	59.3	58.6	56.3
SKNN	51.9	79.0	45.3	54.3	40.0	52.5	60.0	42.5
FINCH	51.4	100.0	46.3	54.1	72.8	59.1	68.6	51.0
SNN	53.4	70.0	43.5	54.3	73.5	38.8	58.6	54.3
CHKNN	54.3	74.0	44.4	55.8	42.1	53.6	67.1	55.8
NTHC	54.3	100.0	46.7	71.5	89.6	44.4	72.9	58.6
SOM	59.6	75.0	48.6	79.0	92.4	65.0	68.5	50.2
BP	55.3	100.0	60.3	68.7	93.6	65.8	82.9	51.2
DAE	55.8	53.0	35.5	55.6	41.7	16.2	32.9	50.3
ClusterGAN	59.6	89.0	39.7	66.4	42.8	29.4	35.7	51.8
SMKNN	59.6	100.0	55.6	81.2	91.5	73.8	66.0	61.4
TRCE	59.6	100.0	52.3	–	89.8	90.1	81.4	–

Table 14. Accuracy comparison the mdK-means algorithm.

Algorithms	Datasets									
	Iris	Glass	Balance	Cancer	Wine	Vertebral	Ecoli	Blood	Seed	Average
KMN	57	36	47	95	90	68	66	53	79	66
HCA	89	46	63	66	40	67	65	76	90	67
FFA	86	48	65	84	70	68	60	76	67	69
CPSO	96	45	37	96	91	68	60	48	89	70
CGA	96	36	41	96	95	70	62	48	90	70
CSMVO	96	52	49	96	96	71	57	48	90	73
DBSACN	68	46	47	93	61	67	43	78	68	63
KMN	85	49	55	96	71	73	66	68	90	73
mdK-means	96	56	61	97	98	78	71	76	94	81
TRCE	95.3	52.3	76.6	96.7	93.3	75.2	78.6	76.2	90.4	81.7

Table 5. The comparison between TRCE and the 5 algorithms presented by Azimi J., *et al*.[10] is shown in Table 15. This table confirmed that the TRCE algorithm never performed the worst. It scored first place in two cases and second place in case. Table 16 summarizes the comparison with the algorithms presented by He Z., *et al*.[50]. The TRCE achieved first place in three cases and second place in one case. More importantly, as presented in Table 17, the TRCE algorithm scored first place in terms of accuracy for the mixed typed datasets which are presented by He Z., *et al*.[47]. Among the seven compared algorithms[47] in Table 18, the TRCE algorithm scored first place in the three datasets. When comparing the TRCE with the clustering algorithms in [26], as presented in Table 19, out of five experimental results, it managed to score first place in three place. Concerning the comparison with the 18 algorithms in [34], as shown in Table 20 the TRCE achieved the first place in 4 cases out of 6 and it never performed the worst. Unlike the compared algorithm, the TRCE algorithm is considered the most practical algorithm as it is not limited to numerical datasets. Table 21 presents the comparison between the REC and the algorithms in [35]. On the 8 datasets, The REC algorithm achieved the first position in four cases and second position in four cases as well, but never performed the worst. The mdK-means algorithm was the most competitive algorithm; however, like most the of compared algorithm, it lacks the ability to handle the categorical and mixed type

Table 15. Ranking of TRCE Algorithm – A.

No.	Datasets	No. of Algorithms	Ranking of TRCE
1	Soybean	5	1
2	Wine		2
3	Iris		1

Table 16. Ranking of TRCE Algorithm - B.

No.	Datasets	No. of Algorithms	Ranking of TRCE
1	Cancer	5	1
2	Votes		1
3	Mushroom		2
4	Zoo	4	1

Table 17. Ranking of TRCE Algorithm – C.

No.	Datasets	No. of Algorithms	Ranking of TRCE
1	Credit Approval	4	1
2	Cleve		1

Table 18. Ranking of TRCE Algorithm - D.

No.	Datasets	No. of Algorithms	Ranking of TRCE
1	Iris	7	1
2	Half-rings		1
3	2-spirals		1

Table 19. Ranking of TRCE Algorithm – E.

No.	Datasets	No. of Algorithms	Ranking of TRCE
1	Cancer	3	1
2	Heart		1
3	Iris		1
4	Votes		4
5	Wine		4

Table 20. Ranking of TRCE Algorithm – F.

No.	Datasets	No. of Algorithms	Ranking of TRCE
1	Sonar	6	1
2	Soybean		1
3	Glass		2
4	Page-Blocks		4
5	Semeion		1
6	SCADI		1

Table 21. Ranking of TRCE Algorithm – G.

No.	Datasets	No. of Algorithms	Ranking of TRCE
1	Iris	6	2
2	Glass		2
3	Balance		1
4	Cancer		2
5	Wine		4
6	Vertebral		2
7	Ecoli		1
8	Blood		1
9	Seed		2

datasets.

To sum up, out of 32 cases, the TRCE algorithm scored advanced positons in the ranking with the first place in 21 cases (62.5%) and second position in 8 cases (25.0%) making a total of 87.5%. Notably, the TRCE algorithm achieved accuracies that were only slightly lower than the method that placed first. The TRCE had the worst performance in two instances; still, its outcomes were competitive. The accuracy for the Vote dataset was 84.6%, whereas for the Wine dataset it was 93.3%.[26] Among all the REC considered is the best algorithm and it is not limited to the numerical dataset with arbitrary shapes. It is capable of handling numerical, categorical and mixed types datasets.

8. Conclusion

This paper presented the framework of the Tri-Level Robust Clustering Ensemble (TRCE) algorithm. The TRCE algorithm seeks to converge the local maximum and it employs a sophisticated method to nominate and select a *BASE*, which helps the algorithm to produce clusters in the same sequence within the interval range. The sophisticated method makes the algorithm to be computationally expensive for large datasets. However, it helps the TRCE to generate reliable clustering results. Besides, it strengthens the algorithm to overcome the issue of cluster indexing. The TRCE algorithm was tested on 19 benchmark datasets and compared with other unsupervised, supervised and ensemble clustering algorithms. The conducted experiments confirmed the strength of the TRCE in producing high clustering results. Among the compared algorithms, TRCE is the absolute winner as it has always been ranked highly. This also confirmed that the TRCE algorithm is accurate, consistent and reliable in general. More importantly, the TRCE algorithm can estimate the most appropriate threshold value and the value of K. Besides, it can handle categorical, numerical, and mixed-type datasets. It has been found that a high value of H does not guarantee high accuracy. Future work should involve improving the efficiency of the proposed ensemble algorithm to be computationally efficient for large datasets.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

References

[1] J. Al-Shaqsi and W. Wang, A novel three staged clustering algorithm, in IADIS European Conference Data Mining, Algarve, Portugal, 2009.

[2] M. Meila, D. Heckerman, An experimental comparison of several clustering and initialization methods, Data Mining and Knowledge Discovery, 1998.

[3] S. Mousavi, F. Z. Boroujeni S. Aryanmehr, Improving customer clustering by optimal selection of cluster centroids in K-means and K-medoids algorithms, *Journal of Theoretical and Applied Information Technology*, 2020, 98, 3807-3814.

[4] M. K. Gupta, P. A. Chandra, Comprehensive survey of data mining, *International Journal of Information Technology*, 2020, **12**, 1243–1257, doi: 10.1007/s41870-020-00427-7.

[5] C. Zhang, W. Huang, T. Niu, Z. Liu, G. Li, D. Cao, Review of clustering technology and its application in coordinating vehicle subsystems, *Automotive Innovation*, 2023, **6**, 89-115, doi: 10.1007/s42154-022-00205-0.

[6] F. Fkih, Similarity measures for collaborative filtering-based recommender systems: review and experimental comparison, *Journal of King Saud University - Computer and Information Sciences*, 2022, **34**, 7645-7669, doi: 10.1016/j.jksuci.2021.09.014.

[7] T. Li, A. Rezaeipanah, E. M. Tag El Din, An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement, *Journal of King Saud University - Computer and Information Sciences*, 2022, **34**, 3828-3842, doi: 10.1016/j.jksuci.2022.04.010.

[8] T. Ma, Z. Zhang, L. Guo, X. Wang, Y. Qian, N. Al-Nabhan, Semi-supervised Selective Clustering Ensemble based on constraint information, *Neurocomputing*, 2021, **462**, 412-425, doi: 10.1016/j.neucom.2021.07.056.

[9] X. Hu, I. Yoo, Cluster ensemble and its applications in gene expression analysis, in Proceedings of the second conference on Asia-Pacific bioinformatics Dunedin, New Zealand 2004.

[10] J. Azimi, S. Davoodi, M. Analoui, Fast Convergence Clustering Ensemble, presented at the 9th International Multi-conference on Information Society, Data Mining and Data Warehouses (SiKDD 2006), Ljubljana, Slovenia, 2006.

[11] D. Greene, A. Tsymbal, N. Bolshakova, P. Cunningham, Ensemble clustering in medical diagnostics, Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems (CBMS'04), June 2004.

[12] A. L. N. Fred, A. K. Jain, Data clustering using evidence accumulation, 2002 International Conference on Pattern Recognition. Quebec City, QC, Canada. IEEE, 2002.

[13] R. Nikbakht, A. Jonsson, A. Lozano, Unsupervised learning for parametric optimization, *IEEE Communications Letters*, 2021, **25**, 678-681, doi: 10.1109/LCOMM.2020.3027981.

[14] A. Rezaeipanah, P. Amiri, H. Nazari, M. Mojarad, H. Parvin, An energy-aware hybrid approach for wireless sensor networks using re-clustering-based multi-hop routing, *Wireless Personal Communications*, 2021, **120**, 3293-3314, doi: 10.1007/s11277-021-08614-w.

[15] A. L. N. Fred, A. K. Jain, Combining multiple clusterings using evidence accumulation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, **27**, 835-850, doi: 10.1109/TPAMI.2005.113.

[16] L. N. F. Ana, A. K. Jain, Robust data clustering, 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. Madison, WI, USA. IEEE, 2003.

[17] J. Ghosh, Multiclassifier systems: Back to the future,

presented at the In F. Roli and J. kittler, editor, Proceeding 3rd International Workshop MCS'02, LNCS 2364, Cagliari, Italy 2003.
[18] X. Z. Fern, C. E. Brodley, Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach, presented at the In Proc. of the 20th International Conference on Machine Learning, Washington DC, USA, 2003.
[19] S. T. Hadjitodorov, L. I. Kuncheva, L. P. Todorova, Moderate diversity for better cluster ensembles, *Information Fusion*, 2006, **7**, 264-275, doi: 10.1016/j.inffus.2005.01.008.
[20] A. L. N. Fred, Finding consistent clusters in data partitions, in 3d International Workshop on Multiple Classifier Systems, 2001.
[21] A. L. N. Fred and A. K. Jain, Data clustering using evidence accumulation, 2002 International Conference on Pattern Recognition, Quebec City, QC, Canada, 2002.
[22] A. Ben-Hur, D. Horn, H. T. Siegelmann, V. Vapnik, A support vector clustering method, *Journal of Machine Learning Research*, 2002, **2**, 125-137, doi: 10.1109/ICPR.2000.906177.
[23] G. J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, 2E. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2008.
[24] P. Domingos, M. Pazzani, On the optimality of the simple bayesian classifier under zero-one loss, *Machine Learning*, 1997, **29**, 103-130, doi: 10.1023/A:1007413511361.
[25] L. I. Kuncheva, S. T. Hadjitodorov, L. P. Todorova, Experimental comparison of cluster ensemble methods, 2006 9th International Conference on Information Fusion. Florence, Italy. IEEE, 2006.
[26] Z. Zhou, G. Si, H. Sun, K. Qu, W. Hou, A robust clustering algorithm based on the identification of core points and KNN kernel density estimation, *Expert Systems with Applications*, 2022, **195**, 116573, doi: 10.1016/j.eswa.2022.116573.
[27] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, *Data Mining and Knowledge Discovery*, 1998, **2**, 283-304, doi: 10.1023/A:1009769707641.
[28] S. Khan , S. Kant, Computation of initial modes for k-modes clustering algorithm using evidence accumulation, in International Joint Conference on Artificial Intelligence Hyderabad, India, 2007.
[29] Z. He, S. Deng, X. Xu, Approximation algorithms for K-modes clustering, Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
[30] Z. He, X. Xu, S. Deng, B. Dong, K-histograms: an efficient clustering algorithm for categorical dataset, arXiv preprint cs/0509033, 2005.
[31] Z. He, X. Xu, S. Deng, TCSOM: clustering transactions using self-organizing map, *Neural Processing Letters*, 2005, **22**, 249-262, doi: 10.1007/s11063-005-8016-3.
[32] S. Aranganayagi, K. Thangavel, Improved K-modes for categorical clustering using weighted dissimilarity measure, World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2009.
[33] Z. He, X. Xu, S. Deng, Scalable algorithms for clustering large datasets with mixed type attributes, *International Journal of Intelligent Systems*, 2005, **20**, 1077-1089, doi: 10.1002/int.20108.
[34] Y. Wang, Y. Ma, H. Huang, B. Wang, D. P. Acharjya, A split–merge clustering algorithm based on the k-nearest neighbor graph, *Information Systems*, 2023, **111**, 102124, doi: 10.1016/j.is.2022.102124.
[35] N. H. M. M. Shrifan, M. F. Akbar, N. A. M. Isa, An adaptive outlier removal aided k-means clustering algorithm, *Journal of King Saud University - Computer and Information Sciences*, 2022, **34**, 6365-6376, doi: 10.1016/j.jksuci.2021.07.003.
[36] C. J. Merz, P. Merphy, UCI Repository of Machine Learning Databases, 1996.
[37] F. D. Bortoloti, E. de Oliveira, P. M. Ciarelli, Supervised kernel density estimation K-means, *Expert Systems with Applications*, 2021, **168**, 114350, doi: 10.1016/j.eswa.2020.114350.
[38] L. N. Fred, K. Jain, Evidence accumulation clustering based on the K-Means algorithm, *Structural, Syntactic, and Statistical Pattern Recognition*, 2002, **2396**, 303-333, doi: 10.1007/3-540-70659-3_46.
[39] K. Jain, M. N. Murty, P. J. Flunn, Data clustering: a review, *ACM Computing Surveys*, 1999, 31, 264-323, doi: 10.1145/331499.331504.
[40] P. Kellam, X. Liu, N. Martin, C. Orengo, S. Swift, A. Tucker, Comparing, Comparing, contrasting and combining clusters in viral gene expression data, in 6th Workshop on Intelligent Data Analysis in Medicine and Pharmocology, 2001.
[41] D. Cristofor, D. A. Simovici, An information-theoretical approach to genetic algorithms for clustering UMass/Boston technical report TR-01-02, 2001.
[42] C. Boulis, M. Ostendorf, Combining multiple clustering systems. European Conference on Principles of Data Mining and Knowledge Discovery. Berlin, Heidelberg: Springer, 2004.
[43] A. Gionis, H. Mannila, P. Tsaparas, Clustering Aggregation, ACM transactions on knowledge discovery from data (TKDD), 2007.
[44] A. Strehl, J. Ghosh, Cluster ensembles - a knowledge reuse framework for combining multiple partitions, *Journal of Machine Learning Research*, 2003, **3**, 583-617.
[45] G. Karypis, V. Kumar, A fast and high-quality multilevel scheme for partitioning irregular graphs, *SIAM Journal on Scientific Computing*, 1998, **20**, 359-392, doi: 10.1137/s1064827595287997.
[46] Z. He, X. Xu, S. Deng, A cluster ensemble method for clustering categorical data, *Information Fusion*, 2005, **6**, 143-151, doi: 10.1016/j.inffus.2004.03.001.
[47] Z. He, X. Xu, S. Deng, Clustering mixed numeric and categorical data: a cluster ensemble approach, ArXiv Computer Science e-prints, 2005.
[48] B. Mobasher, H. Dai, T. Luo, M. Nakagawa, Y. Sun, J. Wiltshire, Discovery and evaluation of aggregate usage profiles for web personalization, *Data Mining and Knowledge Discovery*, 2000, **6**, 61-82, doi: 10.1023/A:1013232803866,
[49] D. Cristofor, D. Simovici, Finding median partitions using

information-theoretical-based genetic algorithms, *Journal of Universal Computer Science*, 2002, **8**, 153–172, doi: 10.3217/jucs-008-02-0153.
[50] Z. He, X. Xu, S. Deng, K-ANMI: a mutual information based clustering algorithm for categorical data, *Information Fusion*, 2008, **9**, 223-233, doi: 10.1016/j.inffus.2006.05.006.
[51] J. Al Shaqsi, W. Wang, Estimating the predominant number of clusters in a dataset, *Intelligent Data Analysis*, 2013, **17**, 603-626, doi: 10.3233/ida-130596.
[52] A. Topchy, A. K. Jain, W. Punch, Combining multiple weak clusterings, Third IEEE International Conference on Data Mining. Melbourne, FL, USA. IEEE, 2003.