ConvexCo: a semi-supervised clustering approach based on adaptive multi-objective Cuckoo in combination with convex hull

A. Taghizabet[a], A. Amini[a,*], J. Tanha[b], and J. Mohammadzadeh[a]

a. Department of Computer Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran
b. Computer and Electrical Engineering Department, Tabriz University, Tabriz, Iran

[*]. Corresponding author.
Email addresses: Email Addresses: a.taghizabet@gmail.com (A. Taghizabet); aamini@kiau.ac.ir (A. Amini); tanha@tabrizu.ac.ir (J. Tanha); j.mohammadzadeh@kiau.ac.ir (J. Mohammadzadeh)

Abstract

Semi-supervised clustering, a technique that combines semi-supervised learning and clustering, is widely employed in the field of machine learning. However, clustering itself poses challenges as it is an NP-hard and multi-objective problem. Consequently, meta-heuristic and multi-objective algorithms have shown greater success in addressing this problem. Nonetheless, these algorithms often encounter issues such as being trapped in local optima and requiring manual parameter adjustments. This research paper introduces an algorithm that tackles the problem of semi-supervised clustering by creating convex hulls of the initial labeled data within each cluster. It also incorporates the labeling of data enclosed within these convex hulls and the adaptive adjustment of parameters using a multi-objective cuckoo algorithm. To enhance the results, labeled data is utilized in the initialization and learning phases of the algorithm. The proposed approach is evaluated using 11 UCI datasets and five synthetic datasets in various experiments. The statistical and numerical analysis demonstrates that the proposed method outperforms the other six algorithms used for comparison. The experiments employ four evaluation criteria, namely ARI, Accuracy, NMI, and F-measure. The results show the superiority of the proposed method across the majority of the datasets.

Keywords: Clustering, Semi-supervised, Convex hull, Adaptive, Swarm Intelligence, Fuzzy adaptation

1. Introduction

The amount of data is increasing every year, and the underlying structures are becoming more and more complex. Therefore, searching, analyzing, and processing this amount of data [1] with such a complex structure presents new challenges that require new data mining and machine learning techniques. The main classifications of machine learning techniques include supervised learning and unsupervised learning [2]. Clustering is considered as one of the most important unsupervised learning techniques in the field of machine learning, and is used in all kinds of applications such as image segmentation [3, 4], self-driving cars [5], identification and analysis of optimal faces in seismic datasets [6], network security issues [7–9], the clustering of sensor nodes [10–13], intrusion detection [14], blind channel equalizer design [14], human action classification [15], document clustering [16], tourism market segmentation [14], analysis of gene expression patterns [17, 18], feature selection [19] etc. as a pre-processing technique.

In some cases, the analyst has little knowledge about the underlying structure of data in the form of prior knowledge (e.g., pairwise constraints and class labels) [20], and the amount of this knowledge compared to the unlabeled data is meager. In such cases, the supervised techniques cannot be used due to the scarcity of training data. On the other hand, unsupervised approaches may also lead to the production of irrelevant results. A better method is to apply the new learning approaches and improve the quality of results by incorporating some prior knowledge in the learning process. These approaches have been produced by combining the supervised and unsupervised learning approaches, and are named semi-supervised learning [21]. Semi-supervised learning can be applied to classification and clustering problems [21, 22].

Compared to the supervised methods, semi-supervised learning algorithm reduce the execution time, and in comparison to the unsupervised methods, they do not get stuck in the local optima [23].

Semi-supervised clustering algorithms are typically classified into three main categories: distance-based, search-based, and hybrid approaches. [24]. In the first methodology, it is common to utilize an existing clustering technique, while incorporating a distance measure based on prior knowledge. The distance criterion is then modified in a manner that it reduces the distance between data points that are intended to be grouped (Must-Link constraints: ML) while increasing the distance between data points that should be assigned to separate clusters (Cannot-Link constraints). [2]. To put it differently, the distance measure is parametrized using the prior knowledge acquired through the ML and Cannot-Link constraints [25]. Nevertheless, in the distance-based approach, the adjusted distance metric may not yield precise results; for instance, two data points linked by a ML constraint could still be distant from each other and consequently assigned to separate clusters. Several studies utilizing this technique include references [26], [27], and [28].

Search-based methodologies adapt conventional clustering algorithms by incorporating prior knowledge, such as labeled data or constraints, to enhance the clustering outcomes. This is achieved through the alteration of the objective function of the clustering algorithm in various manners. The optimization of the clustering objective is achieved by embedding the constraints into the incremental partitioning process in Constrained COBWEB [29]. Seeded K-means [30], on the other hand, incorporates the prior knowledge of the labeled data only during the initialization step of the conventional K-means algorithm. In contrast, Constrained K-means [30]

combines prior knowledge in both the initialization and assignment steps of the K-means algorithm. Combined methods leverage both distance and constraint-based perspectives to effectively address this particular problem.

In the search-based methods, incorporating prior knowledge into traditional clustering techniques enhances clustering performance [2]. One approach utilized in this method involves modeling clustering as a multi-objective optimization problem [2], a strategy that has been implemented in various studies [31–37]. While single-objective methods are also evident in the literature [38–40], the preference for multi-objective algorithms stems from the NP-hard and multi-objective characteristics of the clustering. Multi-objective formulations enhance robustness more significantly than their single-objective counterparts. Both single-objective and multi-objective formulations address challenges such as manual parameter tuning and local optima entrapment. To tackle these issues, this paper introduces ConvexCo, a method that incorporates an adaptive multi-objective cuckoo algorithm with convex hulls to effectively address the semi-supervised clustering problem. The key contributions of this research include the following:

(a) Forming a convex hull in each cluster using labeled data and labeling the enclosed data in it to have more labeled data in the early stages of clustering.

(b) Applying a new definition of penalty component in the calculation of the Connectedness objective function within the multi-objective cuckoo criteria. Exploiting more reliable labeled data to form primary cluster centers.

The subsequent sections of this paper are structured as follows: Section 2 addresses the related work on semi-supervised clustering. Section 3 introduces

the algorithm that has been proposed. Section 4 focuses on the experiments conducted and the results obtained from analyzing the datasets, comparing them with the existing state-of-the-art algorithms. Finally, Section 5 concludes the paper.

2. Related work

In reviewing the performed studies in this area, the relevant papers are categorized into two groups: 1) Papers with multi-objective solutions (Semi-MO group). 2) Papers with single-objective ones (Semi-SO group).

2.1. Semi-MO group

In this section, we review those reports in which multi-objective optimization algorithms have been used to solve the semi-supervised clustering problem.

In the method proposed by Alok et al. [34], four objective functions with the search capability of multi-objective simulated annealing (SA) are simultaneously optimized. The proposed approach aimed to estimate the number of clusters automatically and to detect the appropriate partitioning for the datasets having either well-separated clusters of any shape or symmetrical clusters with or without overlaps. The algorithm has been tested on 24 artificial datasets, 5 UCI datasets, and one satellite image.

In another work, Alok et al. [35] proposed an algorithm to detect intrinsic structures and identify the interesting patterns of five gene expression datasets based on semi-supervised clustering. The methodology of this research is based on the simultaneous optimization of four internal evaluation metrics (Sym-index, I-index, XB-index, and FCM-index) and one external criterion (AR index) using SA multi-objective

optimization algorithm. Due to the difficulty of producing labeled data for gene expression datasets, the FCM algorithm has been used for this purpose in such a way that those data with the highest membership value have been considered as labeled data.

For the segmentation of three satellite images, Alok et al. [33] proposed a semi-supervised algorithm that automatically estimates the number of homogeneous areas using a multi-objective optimization framework. For this reason, the search capability of multi-objective SA has been used, which updates three objective functions (Sym-index, I-index, and Minkowski index) together. The FCM clustering technique has been used to generate supervised data. Labeled data are selected randomly based on the maximum membership values of the respective clusters. The amount of labeled data is ten percent.

For the accurate clustering of three cancer datasets, Saha et al. [32] developed a semi-supervised algorithm to estimate the number of clusters automatically.

Ebrahimi and Abadeh [36] proposed a semi-supervised clustering algorithm for clustering one UCI and three textual datasets. In this algorithm, intra-cluster variance and the number of constraint violations are minimized simultaneously by a multi-objective genetic algorithm.

The goal of the study by Saha et al. [37] was to estimate the number of clusters automatically and also present an appropriate clustering algorithm either in the well-separated partitions of any shape or in symmetrical ones (with or without overlapping). The proposed method has been tested on seven artificial and 4 UCI datasets. Four objective functions of Sym-index (based on symmetry), Con-index (based on cluster connectivity), I-index (based on Euclidean distance), and AR-index

(supervised index) have been optimized simultaneously by multi-objective SA. The first three objective functions are the internal clustering indices, and the last one is an external index.

Khorshidi et al. presented an algorithm for classifying the data collected from patients who have used different health services. The proposed method is based on stochastic approximation of gradient descent optimization of K-median cost and Regression Error functions. This research primarily aimed to evaluate the superiority of the multi-objective method over the single-objective method, to group the patients who had an accident, and to classify new patients [31].

According to the above-reviewed articles, most of the studies in this group have been conducted to solve the automatic clustering problem, and SA is the most widely used optimization algorithm, which demonstrates the need to examine other optimization algorithms and make improvements in other clustering aspects like accuracy and speed.

2.2. Semi-SO group

This section studied the second group of reports. Single-objective optimization algorithms have been used to solve the semi-supervised clustering problem in this group.

Lai et al. [38] proposed an algorithm for the semi-supervised clustering of 13 UCI datasets. The paper aimed to increase the clustering performance of sparsely distributed overlapping clusters by allowing a more informed search using labeled data across a small number of iterations. The proposed algorithm was designed to optimize the ssFCM objective function with the PSO algorithm. The authors used the power of semi-supervised (ssFCM) and PSO methods to conduct a more

informed search using labeled data through fewer iterations while maintaining diversity. The ssFCM can find meaningful clusters using labeled data, with PSO used due to its adaptability to problem representation and proper searching ability. Two methods of ssPSO have been used, where partial supervision is only applied in the initialization phase and the other throughout the learning process.

In the method proposed by Dong et al., a semi-supervised clustering algorithm has been presented to increase the accuracy of fuzzy clustering and solve the problem of the PSO algorithm (being stuck in local optima). The algorithm has been proposed to optimize the reverse of the within-cluster sum of squares in 2 steps: In the first phase, clustering is performed approximately. Then in the second phase, it is conducted more accurately. Four UCI datasets have been used for experimental results [39].

Another semi-supervised clustering algorithm has been presented by Dong et al. The reverse of the sum of the squares of distances between data and their related cluster centers (SSE) has been optimized considering labeled data in combination with the Gini coefficient for increasing clustering accuracy. The algorithm uses the labeled data instead of a random selection of the initial cluster centers. Evaluation of the method has been done using two UCI datasets [40].

In all the reviewed papers, one of the nature-inspired algorithms has been used to solve the problem of semi-supervised clustering in single-objective or multi-objective form. On the other hand, according to [41] and [42], parameter adaptation has a significant effect on establishing a proper balance between the exploration and exploitation phases of metaheuristic algorithms and improves their overall performance. Local traps are also prevented by striking an appropriate balance between the two phases. With this point of view, in this article, an attempt has been

made to dissolve the semi-supervised clustering problem optimally by applying an adaptive multi-objective cuckoo algorithm in combination with the convex hull concept which is formed by each group of the labeled data.

Clustering presents challenges as an NP-hard and multi-objective problem, as discussed earlier [22]. The computational complexity of clustering remains elevated, even for problems of moderate size. Consequently, the utilization of meta-heuristic algorithms, such as swarm intelligence algorithms, has become essential due to their track record of successfully addressing clustering problems. In contrast to a single objective clustering approach, multi-objective clustering seeks to optimize clustering on the basis of multiple criteria simultaneously. Through the utilization of multi-objective optimization algorithms, clustering methods efficiently diminish the scope of search and strive to optimize an array of criteria that are diverse and complementary. The preference for employing these algorithms over single-objective counterparts stems from their remarkable ability to produce resilient outcomes. Our research on swarm intelligence algorithms has provided insights into their inherent traits and capabilities. We concluded that the Cuckoo algorithm has the potential to address the semi-supervised clustering problem according to the following features:

- Strong global search capability in solving many real application optimization problems [43].
- Fewer parameters to adjust [44].
- Ensuring global convergence [45].
- Higher stability of Cuckoo compared to PSO, Bat and Firefly algorithms [46].
- Portability and platform independence [47].
- Producing accurate and high-strength results [48].

Notwithstanding the difficulties related to establishing thresholds and fuzzy rules, the application of fuzzy control techniques remains prevalent due to the promising results they offer. Specifically, these techniques have been seamlessly incorporated into a myriad of swarm intelligence algorithms, including Particle Swarm Optimization (PSO), Bacterial Foraging Optimization (BFO), Ant Colony Optimization (ACO), Artificial Fish Swarm Algorithm (AFSA), Gravitational Search Algorithm (GSA), Firefly Algorithm (FA), Cuckoo Search (CS), Bat Algorithm (BA), and Artificial Bee Colony (ABC). Fuzzy control distinguishes itself among these methods due to its simplicity and effectiveness when incorporated into the PSO algorithm. As a result, it becomes a feasible choice for examination and utilization in alternative algorithms. Therefore, in this paper, the method used in [49] is applied to adapt the migration coefficient parameter in the multi-objective cuckoo algorithm.

3. Proposed semi-supervised Clustering

In this section, we initially define the semi-supervised clustering problem and outline the research goals. Subsequently, we elucidate the sequential process involved in implementing the proposed approach.

3.1. Research objectives and problem definition

Clustering aims to group unlabeled samples into distinct classes based on their similarities, resulting in the formation of clusters [50]. In semi-supervised algorithms, there exists a combination of labeled data $X_l = (x_1, x_2, \dots, x_l)$ with corresponding labels $\{1, \dots, k\}$ and unlabeled data $X_u = (x_{l+1}, x_{l+2}, \dots, x_{l+u})$ with unknown labels, where the number of labeled data points *(l)* is significantly smaller than that of unlabeled data

points *(u)*. Both sets of data (labeled and unlabeled) are selected independently from the same underlying data distribution. The primary objective of semi-supervised techniques is to leverage the available labeled data to enhance the performance of the algorithm and achieve more accurate results.

Our proposed framework designed to solve the semi-supervised clustering problem is illustrated in Figure 1. Given the semi-supervised nature of the clustering problem, we initially aimed to augment the labeled data by constructing a convex hull around them and labeling the data contained in it in each group according to our previous work [51]. Subsequently, after exploring various techniques, we determined that meta-heuristic algorithms are better suited for solving this NP-hard clustering problem. Additionally, since clustering is a multi-objective problem, it is essential to consider different and complementary objectives simultaneously to narrow down the search space and obtain improved results. Hence, we utilized a multi-objective cuckoo algorithm. Furthermore, the initial labeled data were utilized throughout the learning process of the algorithm by customizing the objective functions. It is important to note that only the primary labeled data, rather than the labeled data obtained through convex hull formation, were used in the initialization of the cluster centers. To enhance the overall performance of these algorithms, we thoroughly examined the challenges and practical factors associated with them. Some of these challenges include getting trapped in local optima and the need for manual parameter adjustment. To address these issues, we studied the existing methods for automatic parameter adaptation and implemented the approach described in [49] to dynamically adjust the motion coefficient parameter of the multi-objective cuckoo algorithm.

3.2. Encoding of a state, initialization of clusters centers, and assignment of points

In the presented approach, a collection of real numbers is utilized to denote the condition of Cuckoos. These values signify the positions of the cluster centers and are initially selected from labeled data. In terms of the task at hand, the principle of minimum Euclidean distance has been taken into account for the allocation of unlabeled data. A specific point *P* (as defined in Equations 1 and 2) is assigned to the cluster with the smallest Euclidean distance from its center.

$$P \in i \mid \arg\min_i \{d_e(C_i, P)\} \tag{1}$$

$$i = 1...k \tag{2}$$

C_i is the center of the ith cluster and d_e denotes the Euclidean distance.

3.3. Objective functions

Based on the study by [52], the most suitable objectives for addressing bi-objective clustering problems are Compactness and Connectedness. In our research, we have incorporated these objective functions with certain modifications. The first objective function evaluates the Sum of Squares of Errors (*SSE*) of a solution using Equation 3, which signifies the proximity between an object and the nearest cluster centroid. In cases where labeled data is available, the distance between that data and its assigned cluster is taken into account.

$$SSE(¢) = \Sigma_{i=1}^{k} \Sigma_{\forall x_i \in c_i} \| x_j - \mu_i \|^2 \tag{3}$$

The Euclidean distance is represented by ||.||, with μ_i denoting the center of the cluster

c_i, and x_j representing the j_{th} element of the dataset. The goal is to minimize the *SSE* as the objective function. Additionally, the Connectedness objective is minimized as well, the computation of which is illustrated in Equations 4 and 5.

(4)

$$x_{(p,q)} = \begin{cases} \frac{1}{j} & if \nexists C_k : p \in C_k \wedge q \in C_k \\ 0 & otherwise \end{cases} \quad (5)$$

λ is the number of elements in the clustered dataset, nn_{ij} denotes the j_{th} nearest neighbor of the object *i*, and α indicates the number of neighbors used in the connectedness measure. In addition, $x_{i,nn_{ij}}$ corresponds to the penalty variable value injected as follows: if an object i and its nearest neighbor are not in the same cluster, then the value of the injected penalty is ($1/j$), otherwise it is zero. The nearer the neighbor clustered in a different group, the higher the penalty cost is.

In the customized version of Connectedness, if any labeled data exists in the neighborhood, data comparison is based on the labeled data to impose the penalty, and the cost calculation of the neighbors is done according to the procedure discussed above. In situations where the object i itself (not its neighbors) has a different label, its cost is calculated according to Equation 6:

$$x_{p,q} = \frac{1}{\min(index\ of\ labeled\ data\ in\ ith\ data\ neighborhood)} \quad (6)$$

In other cases, the cost is calculated at $1/j$.

3.4. Adjustments for fuzzy parameter adaptation

In this section, the adjustments used for the Mamdani fuzzy system for the adaptation of the migration coefficient parameter are discussed.

The motion of cuckoos in each generation can be expressed as Equation 7:

$$H_{ij}^{new} = C1 * H_{ij}^{old} + C2 * F * \lambda (H_{ij}^{best} - H_{ij}^{old}) \quad (7)$$

Here, H_{ij}^{new} represents the new position $C1 = (1 - \exp(-\frac{1}{current\,iteration}))$, H_{ij}^{old} denotes the current cuckoo position $C2 = (0.5 + 1.5 * (1/sqrt(currentiteration)))^{Norm(NormalizedError)}$, $Norm(NormalizedError)$ will be explained later, F is the migration coefficient, λ is also a uniform random number between zero and one, and H_{ij}^{best} indicates the position of the best cuckoo in the best cluster in each generation. The migration coefficient F requires manual adjustment; however, we have automated this adjustment using fuzzy logic. The recommended values for this parameter typically fall within the range of 0.5 to 2.5.

To dynamically adapt the migration coefficient parameter, the diversity of the swarm, the error, and the algorithm iterations are utilized as inputs in a fuzzy system framework known as Mamdani. The iteration, diversity, and error are mathematically defined in Equations (8) to (11).

$$Iteration = \frac{Current\,iteration}{Maximum\,of\,iterations} \quad (8)$$

$$Diversity = \frac{Dist(Each\,Cuckoo\ from\ Best\,Cuckoo\,of\,the\,Best\,Cluster)}{Maximum\,Dist} \quad (9)$$

$$Error = \left| Fitness(x_i) - Fitness(BestCuckoo) \right| \quad (10)$$

To ensure that the error value falls within the range of 0 and 1, normalization is performed as per Equation 11.

$$NormalizedError = \frac{Error - \min(Error)}{\max(Error) - \min(Error)} \quad (11)$$

Given the multi-objective nature of the proposed approach, the error vector is represented as a two-dimensional vector. To simplify the calculation and comparison process, the norm form of Equation 11 is utilized as the input for the fuzzy system, as shown in Equation 12.

$$Norm(NormalizedError) = Norm2(NormalizedError) \quad (12)$$

To develop a fuzzy system that can dynamically adjust the migration coefficient parameter, three inputs are taken into consideration. The output of the fuzzy system corresponds to the adjusted migration coefficient. The input variables are structured based on the criteria illustrated in Figure 2(a), 2(b), and 2(c), representing iteration, distance-based diversity, and error, respectively. Each input is divided into three Gaussian membership functions. As previously mentioned, the output variable is constrained within the range of 0.5 to 2.5, defined by five Gaussian membership functions, as depicted in Figure 2(d).

The rules in this fuzzy system are modified based on a specific logic. Initially, the emphasis is on high exploration during the early iterations. As the algorithm progresses towards the final iterations, the focus shifts towards increased exploitation. This adjustment aims to enrich the exploration process and ultimately lead to an almost optimal solution. Conversely, when the population's diversity is low, the cuckoos tend to have minimal dispersion from the best cuckoo and are closely clustered to the best cuckoo. In such cases, it is important to consider increasing the exploration factor. On the other hand, in scenarios with high diversity, the exploitation factor should be prioritized. The specific rules governing the utilized fuzzy system can be observed in Figure 3.

3.5. The ConvexCo algorithm

The steps of the ConvexCo algorithm, which is an adaptive multi-objective cuckoo algorithm for semi-supervised data clustering, are as follows: Upon receiving the inputs, the initial population of cuckoos is created from the labeled data, and the *Goalpoint* (the position of the initial best cuckoo in the best group of cuckoos) is randomly assigned based on the provided pseudo-code. The position of the best cuckoo in all generations is then adjusted according to the initial *Goalpoint*. Subsequently, convex hulls are formed from each group of the labeled data, and the data enclosed within these convex hulls are labeled using their corresponding boundary labels. Non-dominated members of the primary population of cuckoos are identified and added to the archive. The archive is then processed, and the following steps are repeated *MaxIt* times:

Archive members and indices are updated, and the capacity of the archive is checked for additional members. Region-based selection is employed to remove these additional members, with a preference for selecting dense cells for removal. If a selected cell contains only one member, that member is removed; otherwise, one member from the cell is randomly selected for removal.

The best cuckoo from each generation is selected from the archive, with a bias towards selecting from less dense cells. This selection process is reversed when compared to removing additional archive members. The *Globalbest* variable is updated by comparing this selection with its current value. The cuckoo population is then grouped using the K-means algorithm (based on city block distance measurement), and the average values of the objective functions for each group of cuckoos are calculated. These values are then compared to determine the best cluster,

which is defined as a group with a non-dominated response. If no group dominates any others, a group is randomly selected as the best cluster. In that best cluster, a non-dominated cuckoo is also chosen as the best cuckoo. If this non-dominated member does not exist, one cluster member is randomly considered the best cuckoo. This best cuckoo position is the point to which other cuckoos will move in the next step.

For the movement of cuckoos, according to the basic algorithm, the migration coefficient parameter value must be set manually, and due to the adaptive adjustment of this value through a fuzzy system design, at this stage, the input values of the fuzzy system are obtained as described previously in Section 3-4. These input values are then transferred to the fuzzy system to receive the adjusted migration coefficient value corresponding to each iteration. The cuckoos of each generation can move accordingly. After the cuckoo movement, if the iteration number has not reached its maximum value, the actions are resumed from step 10. Otherwise, the algorithm is terminated.

Algorithm 1 ConvexCo

Inputs: *L*: Labeled data; *U*: Unlabeled data; *K*: no. of Clusters; *MaxIt*: Maximum no. of iteration;

Output: Clustered data

1. Use *L* to initialize the cuckoo population
2. Define *Goalpoint* (status of initial best cuckoo in best cuckoo groups) randomly and set *Globalbest* (status of best cuckoo in all generations) with *Goalpoint*
3. Labeling the data enclosed in each convex hull formed from the labeled data in each cluster
4. **For** *It*=1:*MaxIt*
5. **If** (*It*==1)
6. Find non-dominated solutions for the cuckoo population
7. Add non-dominated solutions to the *Archive*
8. Process the *Archive* to select the best solution
9. **end**
10. Update *Archive* elements
11. Update *Archive* indices
12. Check Archive capacity and remove additional solutions

13. Define the best cuckoo of the current generation using the *Archive*
14. Update *Globalbest* by comparing the current best cuckoo and *Globalbest*
15. Group cuckoo population by K-means using 'city block distance'
16. Define the best cuckoo group and the best cuckoo in that group
17. Preparing Fuzzy system entries (*Iteration, Diversity, Error*) for estimating motion coefficient
18. Moving cuckoos toward the best cuckoo
19. **End**

4. Experimental results

Within this segment, we have examined the experiments carried out to validate the effectiveness of the proposed ConvexCo clustering technique. Both numerical and statistical analyses have been conducted on the results obtained. The partitioning outcomes have been compared to other state-of-the-art algorithms, Cuckoo [53], Semi-supervised Cuckoo, NSGAII, MOPSO, Seeded K-means [30], and Constrained K-means [30].

4.1. Parameter setting

The parameter settings of the ConvexCo for the experimental study are presented in Table 1.

4.2. Synthetic data

The synthetic datasets have been denoted in the Xd-Xc-noX format, where 'd' signifies attributes, 'c' represents clusters, and 'no' indicates the dataset number (Table 2). For instance, 2d-10c-no0 refers to a dataset with two attributes, ten clusters, and zero as the dataset number. In order to ensure a fair comparison with other algorithms, all the semi-supervised clustering algorithms in our experiment settings utilize the same labeled dataset. Specifically, 10% of the samples are selected from the labeled data.

Additionally, the number of clusters *'K'* is set to be equal to the number of ground truth clusters.

In this experiment, we clearly demonstrate how the suggested algorithm enhances performance in comparison to alternative algorithms. This improvement is achieved by applying the algorithm on the synthetic dataset 2d-4c-no3. The dataset, along with the labeled data, is depicted in Figure 4(a). Furthermore, Figure 4(b) demonstrates the clustering outcome obtained through the application of the proposed method to the aforementioned dataset. Figure 4(c), on the other hand, represents the desired clustering result. Lastly, Figure 4(d) provides a comparison of the error rate between the proposed method and other algorithms. It is worth noting that the proposed algorithm successfully clusters a significant portion of the unlabeled data while simultaneously reducing the error rate. The reported error rate is based on the average of 15 iterations of executing semi-supervised algorithms.

4.3. Datasets

Table 3 summarizes the characteristics of the 11 benchmark datasets selected from the UCI for the experiment.

4.4. Experimental setup

A 10% portion of the data is extracted from labeled samples for every dataset. The identical sets of the labeled data are utilized in all semi-supervised clustering algorithms. As a result of the stochastic nature of meta-heuristic algorithms, the mean values of *ARI, Accuracy, NMI, and F-measure* are documented in Table 4 following 15 iterations of each algorithm in every dataset.

4.5. Results

Table 4 and its continuation show the significant effectiveness of the proposed algorithm compared to other algorithms under the evaluation criteria of *ARI, Accuracy*, *NMI*, and *F-measure*. The first two columns of each table represent the evaluation values of the Cuckoo clustering algorithm under two distinct criteria. The third to fourth, fifth to sixth, seventh to eighth, ninth to tenth, eleventh to twelfth, and thirteenth to fourteenth columns, respectively, denote the same values for Semi-Cuckoo, NSGAII, MOPSO, Seeded K-means, Constrained K-means, and the proposed ConvexCo method. The clustering algorithm that exhibits the highest performance for each criterion is highlighted in bold for every dataset.

4.5.1. Comparisons to related algorithms

The results in Table 4 show that the ConvexCo algorithm has better clustering performance for most of the datasets compared to the others. These results are more

clearly demonstrated by utilizing the *t-test*.

According to the four evaluation criteria, the performance of the proposed method in the Zoo, Iris, BankAuthentication, Balance, Jain, msplice, Landsat, 2d-10c-no0, 2d-4c-no1, 2d-4c-no2, 2d-4c-no3 datasets is better than that of other methods. Constrained k-means produces better results in Ecoli, Segment, and Pendigits. On Aggregation and 2d-4c-no4, NSGAII works better. Considering the excessive clutter in these five datasets, it seems that the proposed method mislabels the cluttered data due to forming a convex hull at the beginning stage of the algorithm. On the other hand, according to the algorithm routine, these wrong labels are preserved and do not enter the evaluation process by the objective functions again. Another point, compared to the Constrained K-means algorithm, is that this algorithm uses only one criterion for evaluation and the Connectedness measure does not play a role in its evaluation. This criterion is effective in mislabeling due to distance dependence.

The ConvexCo algorithm demonstrates superior effectiveness compared to other semi-supervised algorithms by utilizing labeled data through the formation of convex hulls effectively. Additionally, it optimizes the use of labeled data during the learning phase. Moreover, the fuzzy adaptation of the migration coefficient parameter of cuckoo effectively steers the algorithm in the right direction.

4.5.2. Statistical analysis

In this section, the outcomes derived from Table 4 are examined utilizing *paired t-tests*. Tables 5 to 8 represent the average rank of each algorithm applied to 16 datasets, respectively, based on *ARI*, *Accuracy*, *NMI*, and *F-measure* metrics. According to these rankings, we performed a *paired t-test* for each evaluation criterion between the proposed algorithm and the 3 highest-rank algorithms.

As results presented in Table 9 show, there is a significant difference in the mean of clustering criteria of the ConvexCo algorithm in comparison with those of other comparable algorithms, which rejects the null hypothesis (the mean difference in each group is zero), at $\alpha=0.05$ significance level for each criterion.

[*]*MD* = mean difference, *SD* = standard error of the difference between the means. According to the significance criterion: $p > 0.05$ is not significant.

4.6. Discussion

In most instances, our proposed algorithm has demonstrated superior performance compared to other algorithms. As indicated in Table 4, this is evident from the evaluation criteria, namely *ARI*, *Accuracy*, *NMI*, and *F-measure*. It is important to note that there are certain cases where the overall performance of all algorithms remains low. We have examined these scenarios and presented the findings accordingly.

The datasets we have plotted reveal an interesting pattern. As depicted in Figure 5, these datasets are characterized by a significant overlap of data points. Consequently, distinguishing and assigning them to distinct clusters poses a considerable challenge. Conversely, the Zoo, Iris, 2d-4c-no3, and 2d-4c-no4 datasets exhibit well-separated clusters. As a result, clustering has been performed with greater accuracy, leading to improved evaluation results. This can be observed in Figure 6.

5. Conclusion

In this paper, we applied an adaptive semi-supervised clustering algorithm based on the multi-objective cuckoo named ConvexCo for solving a semi-supervised clustering problem. To reach this aim, due to the multi-objective and NP-hard nature of the clustering, we have used an adaptive multi-objective cuckoo in combination with the convex hull concept in such a way that labeled

data were applied in the initialization step. Then, they participated in forming convex hulls to label the enclosed data in them. Finally, they were exploited in the evaluation step as follows: in the *SSE* criterion, the distance of the labeled data from the centers of the clusters to which they belonged was included, and in the connectedness criterion, if there is any labeled data in the neighborhood, that labeled data is used as a comparison source. To apply a penalty, if the original data is not in the same cluster as the labeled data, it will be penalized to the maximum extent.

The performance of the proposed method was evaluated on several UCI and artificial datasets using four criteria: *ARI*, *Accuracy*, *NMI*, and *F-measure*. According to the experimental results, it is concluded that the proposed algorithm works better than the other compared algorithms in terms of all four criteria.

Future works can be formed by turning single-objective swarm intelligence algorithms that have not been used to solve semi-supervised clustering problems into multi-objective versions and the employment of the ConvexCo objective functions in the other alternative algorithms.

The ConvexCo algorithm may not work well for mixed datasets because of mislabeling data by convex hull formation at the beginning stage of the algorithm and preserving them in the evaluation step. the exploited objective functions in this algorithm are also based on distance which do not operate appropriately on cluttered data. These issues should be deeply considered regarding the clustering of this data type.

Ethics declarations

Conflicts of interest/Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Availability of data and material

The datasets used in the article are well-known data that are currently available to the public.

Code availability

The code will be made available when the paper published.

Consent for publication

The authors declare that they all provide consent for publication.

Contributions

AT, AA: Conceptualization, methodology, software, validation, investigation, resources, visualization, analysis, writing—original draft, writing—review and editing. JT: Conceptualization, methodology, supervision. JM: Conceptualization, review and editing.

References

1. Taha, K. “Semi-supervised and un-supervised clustering: A review and experimental evaluation” , *Inf. Syst.*, **114**, pp. 102-178 (2023).
   DOI: 10.1016/j.is.2023.102178
2. Dinler, D., Tural, M.K. “A survey of constrained clustering” In *Unsupervised Learning Algorithms,* Celebi, M. and Aydin, K., Eds., Springer, Cham (2016).
   DOI: 10.1007/978-3-319-24211-8_9
3. Zhang, H., Li, H., Chen, N., et al. “Novel fuzzy clustering algorithm with variable multi-pixel fitting spatial information for image segmentation” , *Pattern Recognit.*, **121**, pp. 108–201 (2022).
   DOI: 10.1016/J.PATCOG.2021.108201
4. Zhao, F., Cao, L., Liu, H., et al. “Particle competitive mechanism-based multi-objective rough clustering algorithm for image segmentation” , *IEEE Trans. Fuzzy Syst.*, **30**(10), pp. 4127–4141 (2022).
   DOI: 10.1109/TFUZZ.2022.3141752
5. Naeem, S., Ali, A., Anam, S., et al. “An unsupervised machine learning algorithms: comprehensive review”, *Int. J. Comput. Digit. Syst*., **13**(1), pp. 911–921 (2023).
   DOI: 10.12785/ijcds/130172
6. Hashemi, H., De Beukelaar, P., Beiranvand, B., et al. “Clustering seismic datasets for optimized facies analysis using a SSCSOM technique”, *79th EAGE Conference and Exhibition*, Paris, France (2017).
   DOI: 10.3997/2214-4609.201700916
7. Ma, X., Keung, J., Yang, Z., et al. “CASMS: Combining clustering with attention semantic model for identifying security bug reports”, *Inf. Softw. Technol.*, **147** pp. 106–906 (2022).
   DOI: 10.1016/j.infsof.2022.106906
8. Ye, W., Wang, H., Zhong, Y. “Optimization of network security protection situation based on data clustering”, *Int. J. Syst. Assur. Eng. Manag*., pp. 1–8 (2022).
   DOI: 10.1007/s13198-021-01529-6
9. Kanthimathi, N., Roshini Roy, J., Saranya, N., et al. “Trust-based security scheme using fuzzy clustering for vehicular ad hoc networks”, In *Advances in Intelligent Systems and Computing Ranganathan,* G., Fernando, X., Shi, F., and El Allioui, Y., Eds., Springer, Singapore (2022).
   DOI: 10.1007/978-981-16-5301-8_32

10. Sathyamoorthy, M., Kuppusamy, S., Dhanaraj, R.K., et al. “Improved K-means based q learning algorithm for optimal clustering and node balancing in WSN”, *Wirel. Pers. Commun.*, **122**(3), pp. 2745–2766 (2021).
DOI: 10.1007/s11277-021-09028-4
11. Sharma, R., Vashisht, V., Singh, U. “eeFFA/DE - A fuzzy-based clustering algorithm using hybrid technique for wireless sensor networks” , *Int. J. Adv. Intell. Paradig*., **21**(1–2), pp. 129–157 (2022).
DOI: 10.1504/IJAIP.2022.121034
12. Jayaraman, G., Dhulipala, V.R.S. “FEECS: Fuzzy-based energy-efficient cluster head selection algorithm for lifetime enhancement of wireless sensor networks.” , *Arab. J. Sci. Eng.*, **47**(2), pp. 1631–1641 (2021).
DOI: 10.1007/s13369-021-06030-7
13. Srinivas, M., Amgoth, T. “Data acquisition in large-scale wireless sensor networks using multiple mobile sinks: a hierarchical clustering approach” , *Wirel. Networks*, **28**(2), pp. 603–619 (2022).
DOI: 10.1007/s11276-021-02845-2
14. Ezugwu, A.E., Ikotun, A.M., Oyelade, O.O., et al. “A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects” , *Eng. Appl. Artif. Intell.*, **110**, pp. 104–743 (2022).
DOI: 10.1016/j.engappai.2022.104743
15. Sun, R. “A recognition method for visual image of sports video based on fuzzy clustering algorithm” , *Int. J. Inf. Commun. Technol.*, **20**(1), pp. 1–17 (2022).
DOI: 10.1504/IJICT.2022.119311
16. Sardar, T.H., Ansari, Z. “MapReduce-based fuzzy c-means algorithm for distributed document clustering” , *J. Inst. Eng. Ser. B*, **103**(1), pp. 131–142 (2021).
DOI: 10.1007/s40031-021-00651-0
17. Agapito, G., Fedele, G. “Clustering methods for microarray datasets” , *Methods Mol. Biol.*, **2401**, pp. 249–261 (2022).
DOI: 10.1007/978-1-0716-1839-4_16
18. Sharma, C.M., Dinkar, S.K. “A survey on evolutionary clustering algorithms and applications” , In *Applications of Advanced Optimization Techniques in Industrial Engineering*, 1st Edn., CRC Press (2022).
DOI: 10.1201/9781003089636-2
19. Zhang, J., Tian, J., Yan, P., et al. “Multi-hop graph pooling adversarial

network for cross-domain remaining useful life prediction: A distributed federated learning perspective" , *Reliab. Eng. Syst. Saf.,* **244**, pp. 109-950 (2024).
DOI: 10.1016/j.ress.2024.109950

20. Cai, J., Hao, J., Yang, H., et al. "A review on semi-supervised clustering" , *Inf. Sci. (Ny).*, **632**, pp. 164–200 (2023).
DOI: 10.1016/j.ins.2023.02.088
21. José-García, A., Gómez-Flores, W. "Automatic clustering using nature-inspired metaheuristics: A survey" , *Appl. Soft Comput. J.*, **41**, pp. 192–213 (2016).
DOI: 10.1016/j.asoc.2015.12.001
22. Engelen, J.E., Hoos, H.H. "A survey on semi-supervised learning" , *Mach. Learn.*, **109**(2), pp. 373–440 (2020).
DOI: 10.1007/s10994-019-05855-6
23. Saha, S., Alok, A., Ekbal, A., et al. "Brain image segmentation using semi-supervised clustering" , *Exp. Sys. With Apps.*, **52**, pp. 50–63 (2016).
DOI: 10.1016/j.eswa.2016.01.005
24. Qin, Y., Ding, S., Wang, L., et al. "Research progress on semi-supervised clustering" , *Cognit. Comput.*, **11**, pp. 599–612 (2019).
DOI: 10.1007/s12559-019-09664-w
25. Nanda, S.J., Panda, G. "A survey on nature inspired metaheuristic algorithms for partitional clustering" , *Swarm Evol. Comput.*, **16**, pp. 1–18 (2014).
DOI: 10.1016/j.swevo.2013.11.003
26. Sanodiya, R.K., Saha, S., Mathew, J. "A kernel semi-supervised distance metric learning with relative distance: Integration with a MOO approach" , *Expert Syst. Appl.*, **125**, pp. 233–248 (2019).
DOI: 10.1016/j.eswa.2018.12.051
27. Zhang, Z., Kwok, J.T., Yeung, D.-Y. "Parametric distance metric learning with label information." *Proc International Joint Conference on Artificial Intelligence* (2003).
28. Baghshah, M., Shouraki, S., 2010, N. "Kernel-based metric learning for semi-supervised clustering" , *Neurocomputing*, **73**(7–9), pp. 1352–1361 (2010).
DOI: 10.1016/j.neucom.2009.12.009
29. Wagstaff, K., Cardie, C. "Clustering with Instance-level Constraints" , *AAAI/IAAI*, **1097**, pp. 577–584 (2000).
30. Basu, S., Banerjee, A., Mooney, R. "Semi-supervised clustering by seeding" , *Proceedings of the 19th International Conference on*

Machine Learning (ICML-2002) , Sydney, Australia (2002), https://www.cs.utexas.edu/~ml/papers/semi-icml-02.pdf.

31. Akbarzadeh Khorshidi, H., Aickelin, U., Haffari, G., et al. "Multi-objective semi-supervised clustering to identify health service patterns for injured patients" , *Heal. Inf. Sci. Syst*., **7**(1), pp. 1–8 (2019). DOI: 10.1007/s13755-019-0080-6
32. Saha, S., Kaushik, K., Alok, A.K., et al. "Multi-objective semi-supervised clustering of tissue samples for cancer diagnosis" , *Soft Comput.*, **20**(9), pp. 3381–3392 (2016). DOI: 10.1007/s00500-015-1783-5
33. Kumar Alok, A., Saha, S., Ekbal, A. "Multi-objective semi-supervised clustering for automatic pixel classification from remote sensing imagery" , *Soft Comput.*, **20**(12), pp. 4733–4751 (2016). DOI: 10.1007/s00500-015-1701-x
34. Alok, A.K., Saha, S., Ekbal, A. "A new semi-supervised clustering technique using multi-objective optimization" , *Appl. Intell.*, **43**(3), pp. 633–661 (2015). DOI: 10.1007/s10489-015-0656-z
35. Kumar Alok, A., Saha, S., Ekbal, A. "Semi-supervised clustering for gene-expression data in multiobjective optimization framework" , *Int. J. Mach. Learn. Cybern.*, **8**(2), pp. 421–439 (2017). DOI: 10.1007/s13042-015-0335-8
36. Ebrahimi, J., Abadeh, M.S. "Semi supervised clustering: a pareto approach." In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, Perner, P., ed., Springer, Berlin, Heidelberg (2012). DOI: 10.1007/978-3-642-31537-4_19
37. Saha, S., Ekbal, A., Alok, A.K. "Semi-supervised clustering using multiobjective optimization" , *Proceedings of the 2012 12th International Conference on Hybrid Intelligent Systems*, IEEE, Pune, India (2012). DOI: 10.1109/HIS.2012.6421361
38. Lai, D.T.C., Miyakawa, M., Sato, Y. "Semi-supervised data clustering using particle swarm optimisation" , *Soft Comput*., **24**(5), pp. 3499–3510 (2020). DOI: 10.1007/s00500-019-04114-z
39. Dong, J., Qi, M., Wang, F. "A two-stage semi-supervised clustering method based on hybrid particle swarm optimization." *1st International Conference on Electronics Instrumentation and Information Systems*, IEEE, Harbin, China (2017).

DOI: 10.1109/EIIS.2017.8298609
40. Dong, J., Qi, M., Wang, F. “An improved artificial bee colony algorithm for solving semi-supervised clustering” , *5th International Conference on Computer Science and Network Technology*, Changchun, China (2016).
DOI: 10.1109/iccsnt.2016.8070171
41. Salgotra, R., Singh, U., Saha, S. “New cuckoo search algorithms with enhanced exploration and exploitation properties” , *Expert Syst. Appl.*, **95**, pp. 384–420 (2017).
DOI: 10.1016/j.eswa.2017.11.044
42. Mlakar, U., Jr, I., Fister, I., et al. “Hybrid self-adaptive cuckoo search for global optimization” , *Swarm and Evolu. Comp.*, **29**, pp. 47–72 (2016).
DOI: 10.1016/j.swevo.2016.03.001
43. Yang, X.S., Deb, S., Mishra, S.K. “Multi-species Cuckoo Search Algorithm for Global Optimization” , *Cognit. Comput.*, **10**(6), pp. 1085–1095 (2018).
DOI: 10.1007/s12559-018-9579-4
44. Goel, S., Sharma, A., Bedi, P., et al. “Cuckoo Search Clustering Algorithm: A novel strategy of biomimicry” , *World Congrss on Information and Communication Technologies,* Mumbai, India, pp. 916-921 (2011).
DOI: 10.1109/WICT.2011.6141370
45. Yang, X.S., Deb, S. “Cuckoo search: Recent advances and applications” , *Neural Compu. & Applic*, **24**, pp. 169–174 (2014).
DOI: 10.1007/s00521-013-1367-1
46. Gong, X., Liu, L., Fong, S., et al. “Comparative research of swarm intelligence clustering algorithms for analyzing medical data” , *IEEE Access*, **7**, pp. 137560–137569 (2019).
DOI: 10.1109/ACCESS.2018.2881020
47. Wang, G. “A Comparative Study of Cuckoo Algorithm and Ant Colony Algorithm in Optimal Path Problems*”, 2nd International Conference on Electronic Information Technology and Computer Engineering,* **232** (2018).
DOI: 10.1051/matecconf/201823203003
48. Garg, S., Batra, S. “Fuzzified cuckoo based clustering technique for network anomaly detection” , *Comp. & Electri. Engi.* , **71**, pp. 798–817 (2018).
DOI: 10.1016/j.compeleceng.2017.07.008
49. Melin, P., Olivas, F., Castillo, O., et al. “Optimal design of fuzzy

classification systems using PSO with dynamic parameter adaptation through fuzzy logic" , *Expert Syst. with Appl.*, **40**(8), pp. 3196–3206 (2013). DOI: 10.1016/J.ESWA.2012.12.033

50. Wei, S., Li, Z., Zhang, C. "A semi-supervised clustering ensemble approach integrated constraint-based and metric-based" *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service,* ACM, New York, NY, USA (2015). DOI: 10.1145/2808492.2808518
51. Taghizabet, A., Tanha, J., Amini, A., et al. "A semi-supervised clustering approach using labeled data" , *Sci. Iran*., **30**(1 D), pp. 104–115 (2023).

DOI: 10.24200/sci.2022.58519.5772

52. Kishor, A., Singh, K., Prakash, J. "NSABC: Non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering" , *Neurocomputing*, **216**, pp. 514-533 (2016). DOI: 10.1016/j.neucom.2016.08.003
53. Rajabioun, R. "Cuckoo optimization algorithm" , *Appl. Soft Comput*., **11**(8), pp. 5508–5518 (2011). DOI: 10.1016/J.ASOC.2011.05.008

Biographies

Atiyeh Taghizabet received B.Sc degree in computer engineering from the Islamic Azad University South Tehran, Iran in 2006. She received her M.Sc degree in software engineering in Payam Noor University, Rey Branch, Iran in 2011. She is currently Ph.D candidate of software systems and her research interests include optimization algorithms, metaheuristic algorithms, swarm intelligence, data mining.

Jafar Tanha received the B.Sc and M.Sc degree in computer science from the university of AmirKabir (Polytechnic), Tehran, Iran, in 1999 and 2001, respectively, and the Ph.D degree in computer science-Artificial Intelligence from the University of Amsterdam (UvA), Amsterdam, The Netherlands, in 2013. He joined at INL institute, Leiden, The Netherland, as a researcher, from 2013 to 2015. Since 2015, he has been with the Department of Computer Engineering, Payame-Noor University, Tehran, Iran, where he was an Assistance Professor. He has held lecturing positions at the Iran university of Science & Technology, Tehran, Iran, in 2016. His current position is the assistant professor at the University of Tabriz, Tabriz, Iran. His main areas of research interest are machine learning, pattern recognition, and document analysis.

Amineh Amini received her B.Sc. degree in software engineering from Mashhad Azad University in 2001. She obtained her M.Sc. degree in the same field from Najafabad Azad University in 2005. Then, she received her Ph.D degree and post-doctoral from University of Malaya. She is a faculty member and head of department of computer engineering of Karaj Azad University. Her main research interests include data mining and software re-modularization.

Javad Mohammadzadeh received his B.Sc degree in computer science from Shahid Bahonar University of Kerman, Iran in 2004. He received his M.Sc degree in computer science from the University of Tehran, in 2007, and Ph.D degree in bioinformatics from the University of Tehran, in 2014. His research interests include swarm intelligence algorithms, bioinformatics algorithms, complex dynamical networks, parallel computing, and deep learning.

Figure 1. The proposed framework to solve the semi-supervised clustering problem.
Figure 2. (a) *Iteration* input.
Figure 2. (b) *Diversity* input.
Figure 2. (c) *NormalizedError* input.
Figure 2. (d) Output (adjusted migration coefficient: *F*).
Figure 3. The designed fuzzy system rules used to estimate the migration coefficient.
Figure 4. (a) Labeled and unlabeled examples.
Figure 4. (b) Clustered examples using ConvexCo.
Figure 4. (c) Original fully labeled data.
Figure 4. (d) Error Rate comparison of the different algorithms.
Figure 5. Datasets with mix clusters: (a) Ecoli.
Figure 5. Datasets with mix clusters: (b) Segment.
Figure 6. Well-separated datasets: (a) Zoo.
Figure 6. Well-separated datasets: (b) Iris.
Figure 6. Well-separated datasets: (c) 2d-4c-no3.
Figure 6. Well-separated datasets: (d) 2d-4c-no4.

Table 1. Parameter settings.
Table 2. Synthetic dataset characteristics.
Table 3. UCI dataset characteristics.
Table 4. Comparing the performance of the proposed method with others using 10% labeled data for semi-supervised algorithms.
Table 4 Continued. Comparing the performance of the proposed method with others using 10% labeled data for semi-supervised algorithms.
Table 5. Statistical rank based on the *ARI* index.
Table 6. Statistical rank based on *Accuracy*.
Table 7. Statistical rank based on *NMI*.
Table 8. Statistical rank based on *F-measure*.
Table 9. Results of the significance testing by *paired t-test* based on *ARI, Acc, NMI,* and *F-measure* criteria.

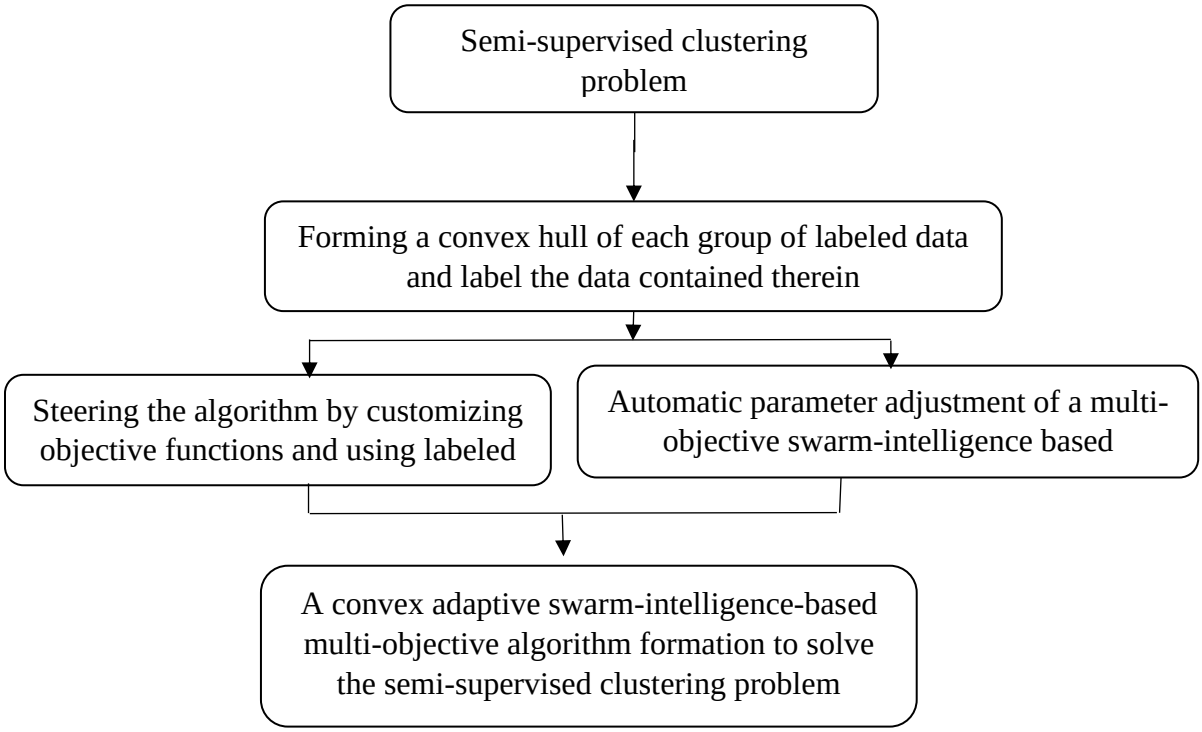
Semi-supervised clustering problem
Forming a convex hull of each group of labeled data and label the data contained therein
Steering the algorithm by customizing objective functions and using labeled
Automatic parameter adjustment of a multi-objective swarm-intelligence based
A convex adaptive swarm-intelligence-based multi-objective algorithm formation to solve the semi-supervised clustering problem

Figure 1.

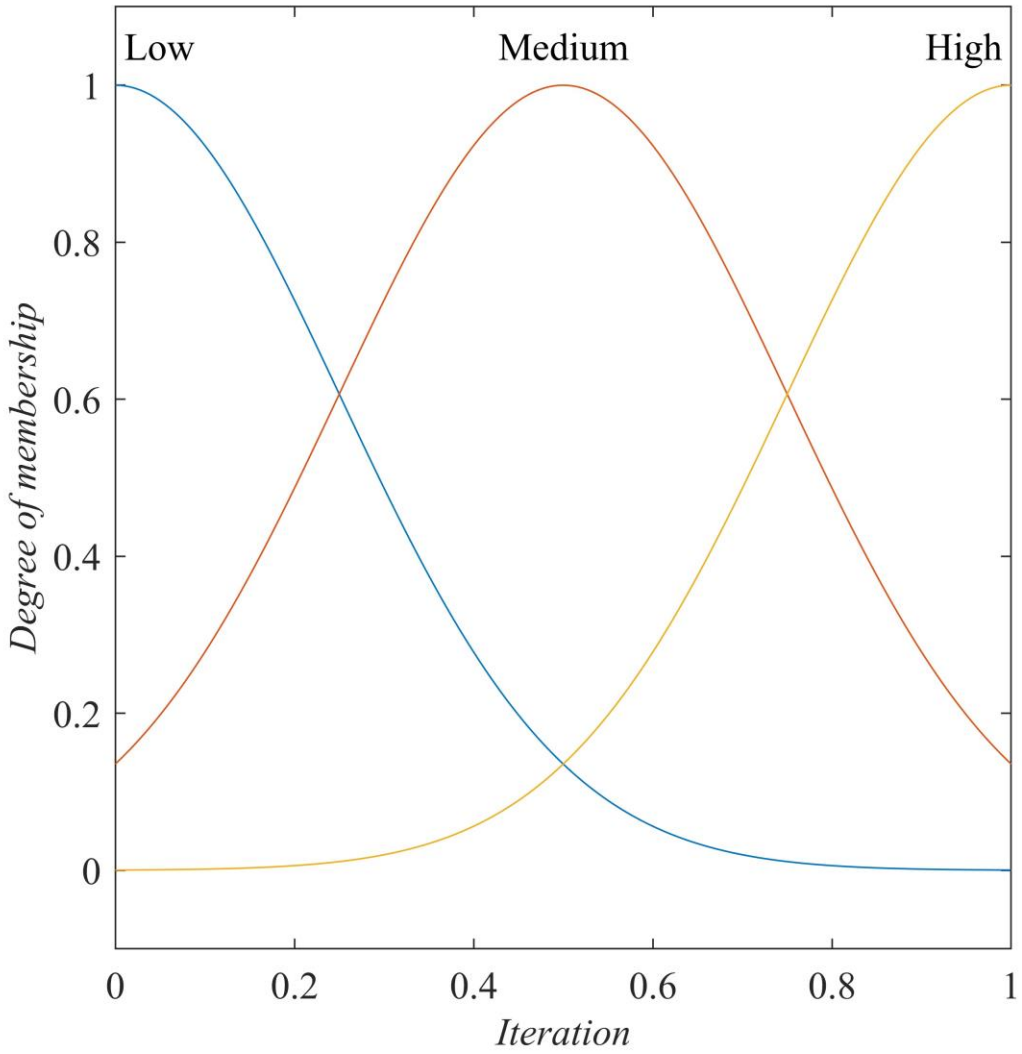
Low
Medium
High
1
0.8
0.6
0.4
0.2
0
Degree of membership
0
0.2
0.4
0.6
0.8
1
Iteration

Figure 2. (a)

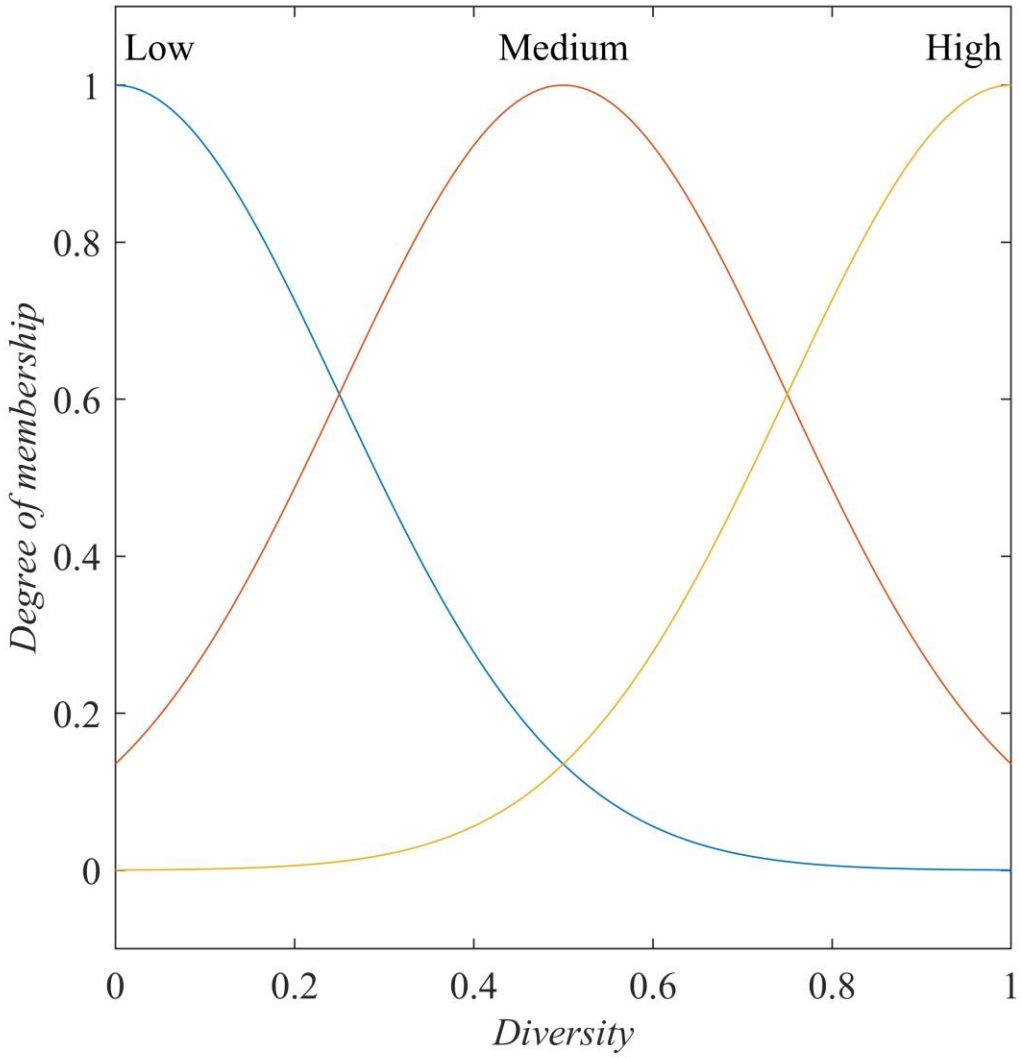
Low
Medium
High
1
0.8
0.6
0.4
0.2
0
Degree of membership
0
0.2
0.4
0.6
0.8
1
Diversity

Figure 2. (b)

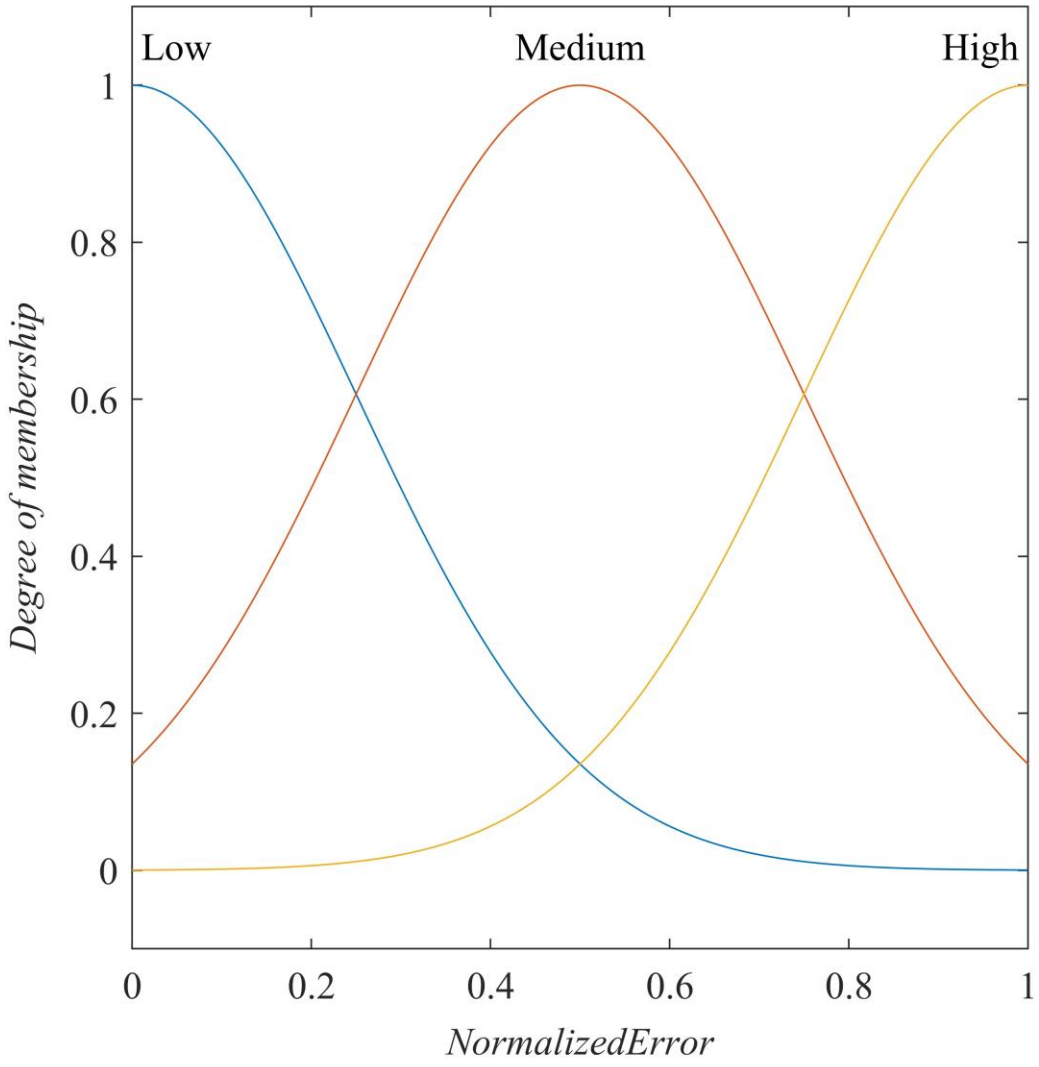
Low
Medium
High
1
0.8
0.6
0.4
0.2
0
Degree of membership
0
0.2
0.4
0.6
0.8
1
NormalizedError

Figure 2. (c)

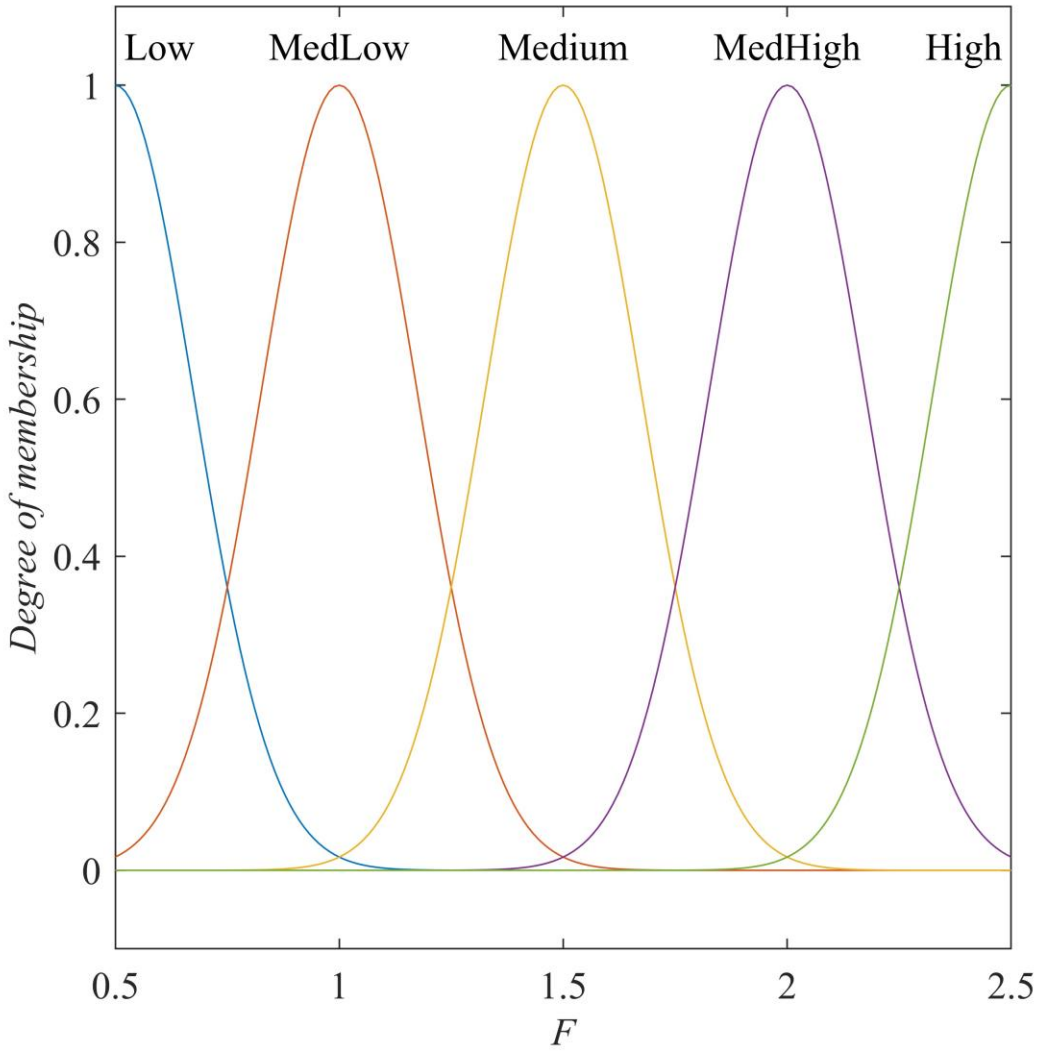
Low
MedLow
Medium
MedHigh
High
1
0.8
0.6
0.4
0.2
0
Degree of membership
0.5
1
1.5
2
2.5
F

Figure 2. (d)

1. If (*Iteration* is Low) and (*Diversity* is Low) and (*NormalizedError* is Low) then (*F* is Low) (1)
2. If (*Iteration* is Low) and (*Diversity* is Low) and (*NormalizedError* is Medium) then (*F* is Low) (1)
3. If (*Iteration* is Low) and (*Diversity* is Low) and (*NormalizedError* is High) then (*F* is Low) (1)
4. If (*Iteration* is Low) and (*Diversity* is Medium) and (*NormalizedError* is Low) then (*F* is Medium) (1)
5. If (*Iteration* is Low) and (*Diversity* is Medium) and (*NormalizedError* is Medium) then (*F* is MedLow) (1)
6. If (*Iteration* is Low) and (*Diversity* is Medium) and (*NormalizedError* is High) then (*F* is Low) (1)
7. If (*Iteration* is Low) and (*Diversity* is High) and (*NormalizedError* is Low) then (*F* is MedLow) (1)
8. If (*Iteration* is Low) and (*Diversity* is High) and (*NormalizedError* is Medium) then (*F* is Medium) (1)
9. If (*Iteration* is Low) and (*Diversity* is High) and (*NormalizedError* is High) then (*F* is Low) (1)
10. If (*Iteration* is Medium) and (*Diversity* is Low) and (*NormalizedError* is Low) then (*F* is Medium) (1)
11. If (*Iteration* is Medium) and (*Diversity* is Low) and (*NormalizedError* is Medium) then (*F* is MedLow) (1)
12. If (*Iteration* is Medium) and (*Diversity* is Low) and (*NormalizedError* is High) then (*F* is Low) (1)
13. If (*Iteration* is Medium) and (*Diversity* is Medium) and (*NormalizedError* is Low) then (*F* is MedHigh) (1)
14. If (*Iteration* is Medium) and (*Diversity* is Medium) and (*NormalizedError* is Medium) then (*F* is Medium) (1)
15. If (*Iteration* is Medium) and (*Diversity* is Medium) and (*NormalizedError* is High) then (*F* is MedLow) (1)
16. If (*Iteration* is Medium) and (*Diversity* is High) and (*NormalizedError* is Low) then (*F* is MedHigh) (1)
17. If (*Iteration* is Medium) and (*Diversity* is High) and (*NormalizedError* is Medium) then (*F* is MedHigh) (1)
18. If (*Iteration* is Medium) and (*Diversity* is High) and (*NormalizedError* is High) then (*F* is Medium) (1)
19. If (*Iteration* is High) and (*Diversity* is Low) and (*NormalizedError* is Low) then (*F* is MedLow) (1)
20. If (*Iteration* is High) and (*Diversity* is Low) and (*NormalizedError* is Medium) then (*F* is Medium) (1)
21. If (*Iteration* is High) and (*Diversity* is Low) and (*NormalizedError* is High) then (*F* is Low) (1)
22. If (*Iteration* is High) and (*Diversity* is Medium) and (*NormalizedError* is Low) then (*F* is MedHigh) (1)
23. If (*Iteration* is High) and (*Diversity* is Medium) and (*NormalizedError* is Medium) then (*F* is MedHigh) (1)
24. If (*Iteration* is High) and (*Diversity* is Medium) and (*NormalizedError* is High) then (*F* is Medium) (1)
25. If (*Iteration* is High) and (*Diversity* is High) and (*NormalizedError* is Low) then (*F* is High) (1)
26. If (*Iteration* is High) and (*Diversity* is High) and (*NormalizedError* is Medium) then (*F* is MedHigh) (1)
27. If (*Iteration* is High) and (*Diversity* is High) and (*NormalizedError* is High) then (*F* is MedLow) (1)

Figure 3.

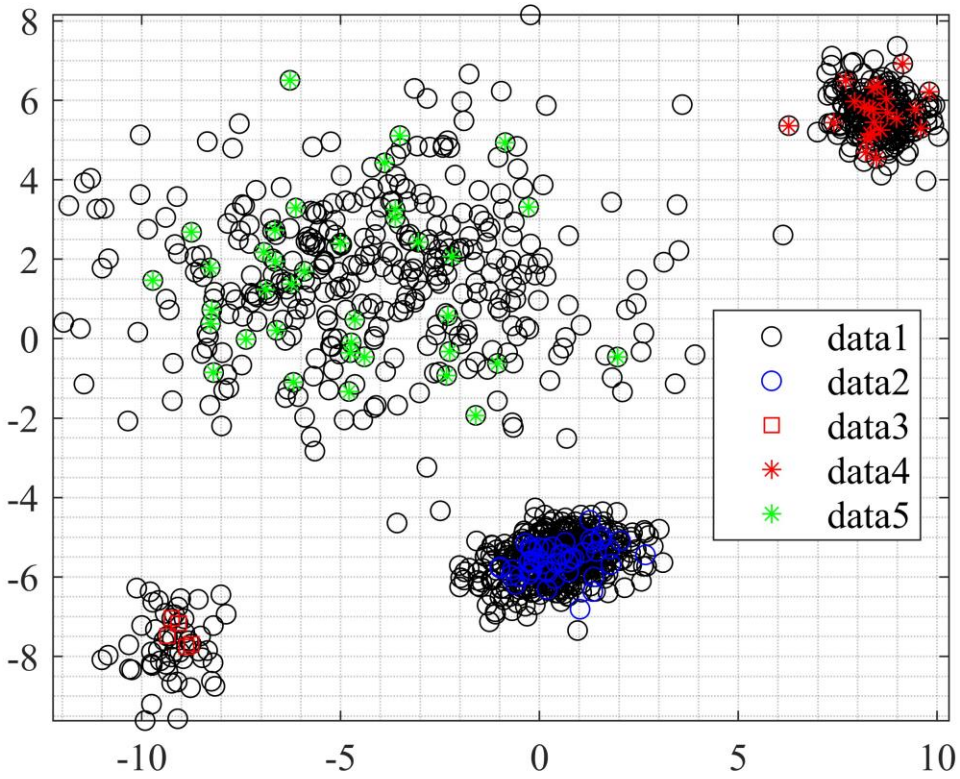

Figure 4. (a)

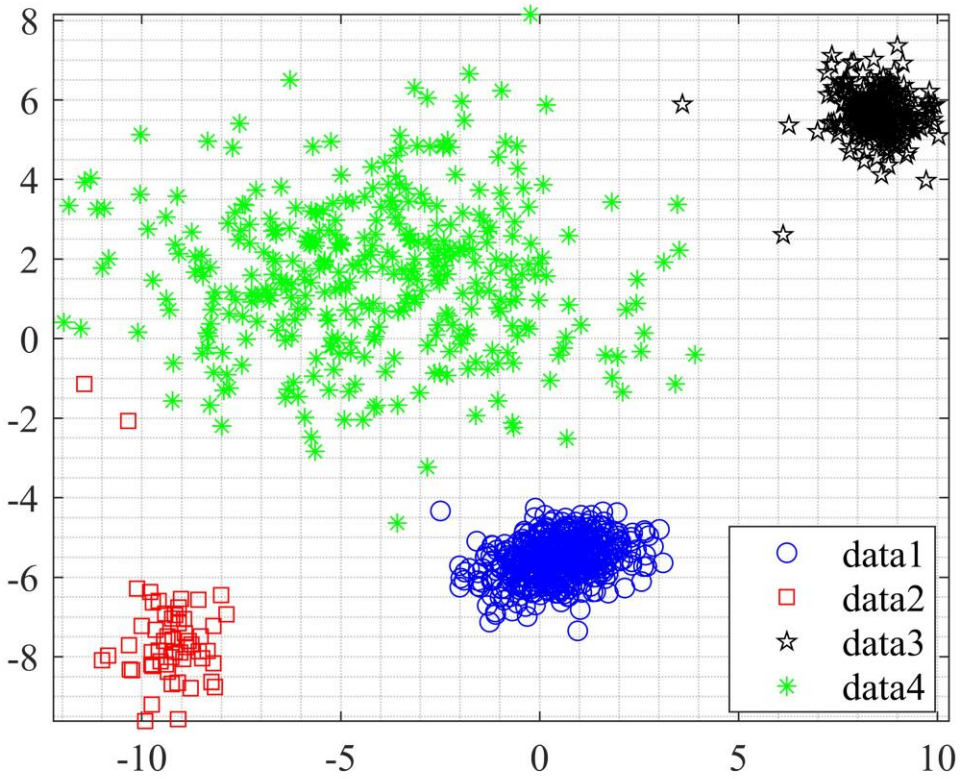

Figure 4. (b)

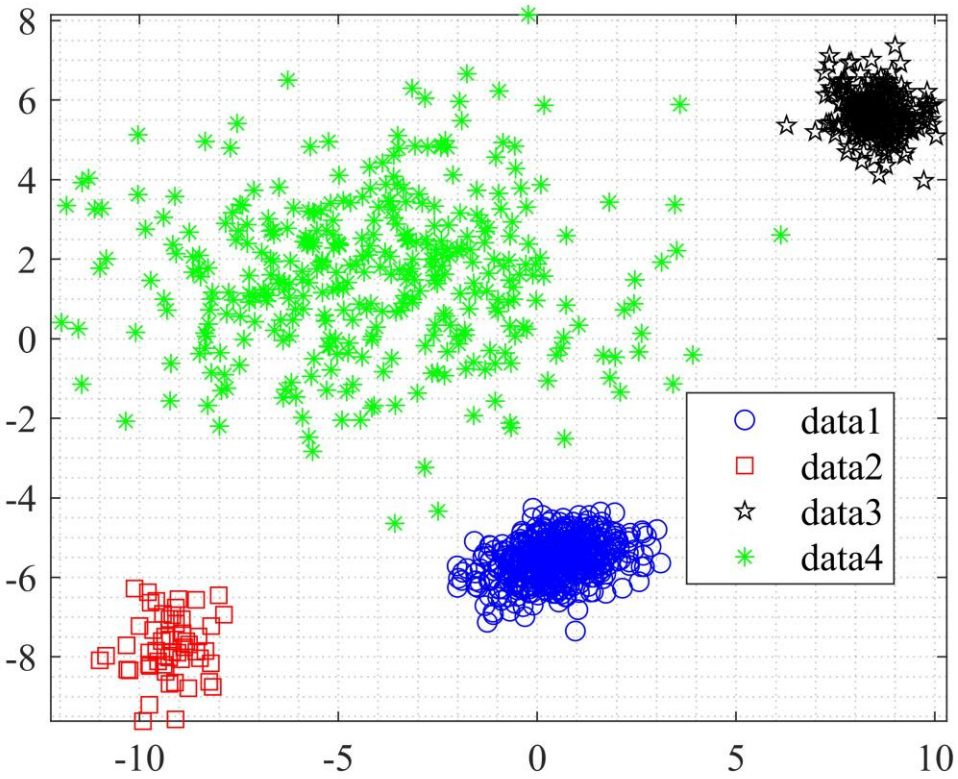

Figure 4. (c)

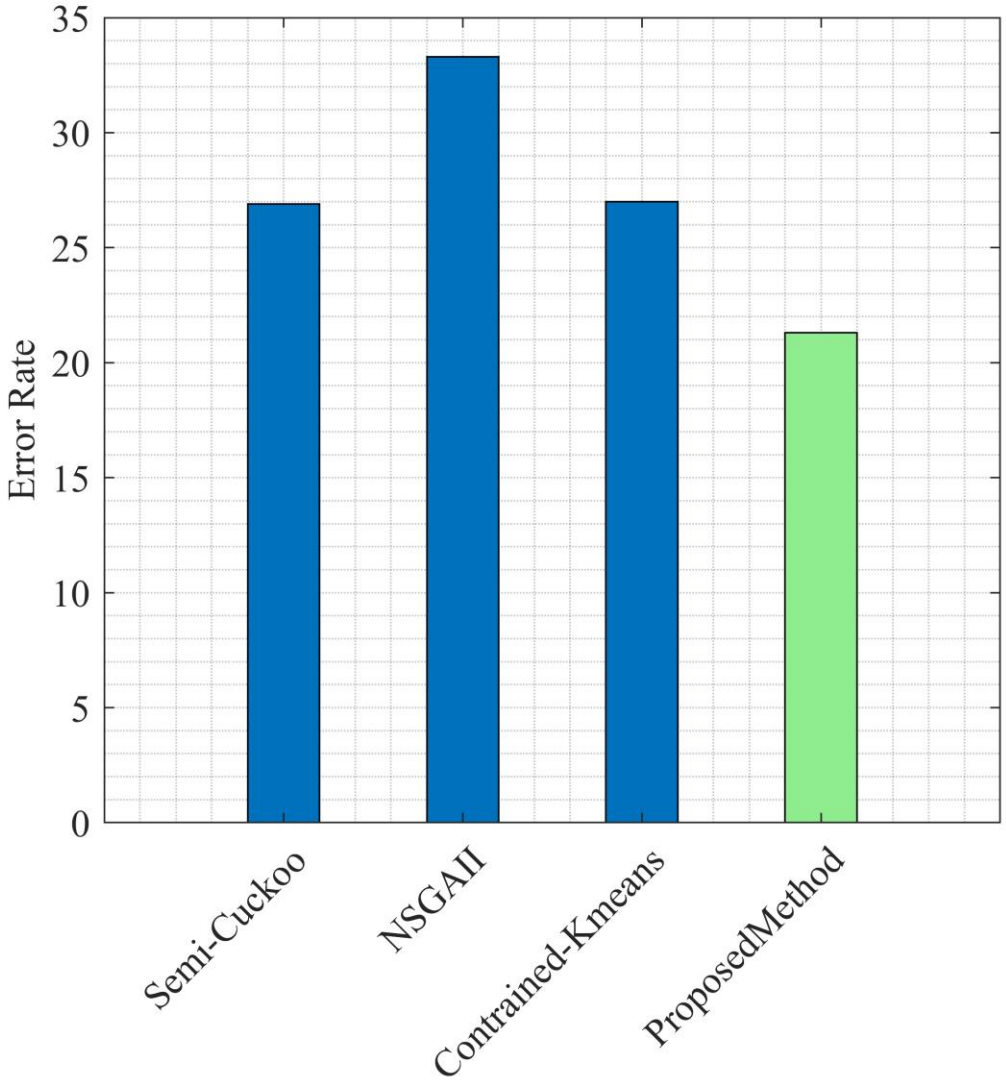
35
30
25
20
15
10
5
0
Error Rate
Semi-Cuckoo
NSGAII
Contrained-Kmeans
ProposedMethod

Figure 4. (d)

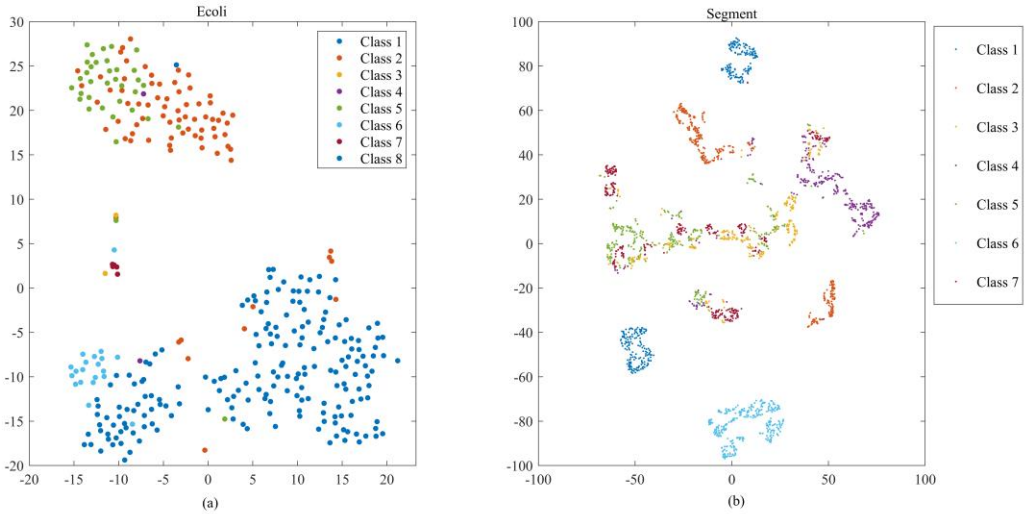

Figure 5.

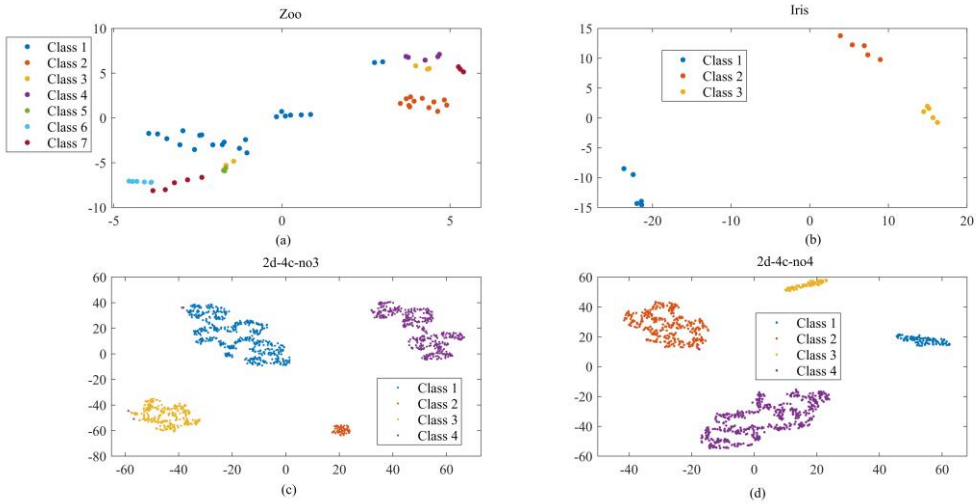

Figure 6.

Parameters	Explanations	Values
K	Number of clusters	Different depending on each dataset
NumCuckoos	Number of initial cuckoos	10
MinNumberOfEggs	The minimum number of eggs that can be laid by each cuckoo	3
MaxNumberOfEggs	The maximum number of eggs which can be laid by each cuckoo	5
NumberOfCluster	The number of clusters in grouping cuckoos by K-means algorithm	3
MaxNumOfCuckoo	Maximum number of cuckoo population	15
MaxIter	Maximum iterations	100

Table 1.

Name	#Example	#Attributed (d)	#Class
2d-10c-no0	2972	2	10
2d-4c-no1	1623	2	4
2d-4c-no2	1064	2	4
2d-4c-no3	1123	2	4
2d-4c-no4	863	2	4

Table 2.

Name	#Example	#Attributed(D)	#Class
Zoo1707	101	17	7
Iris0403	150	4	3
Ecoli0708	336	7	8
Segment1907	2310	19	7
Pendigits1610	10992	16	10
BankAuthentication0402	1372	4	2
Balance 0403	625	4	3
Jain0202	373	2	2
aggregation0207	788	2	7
msplice1003	3175	240	3
landsat1006	2000	36	6

Table 3.

Algorithm / Dataset	Cuckoo		Semi-Cuckoo		NSGAII		MOPSO		Seeded K-means		Constrained K-means		Convex-Co	
	ARI	*Acc*	*ARI*	*Acc*	*ARI*	*Acc*	*ARI*	*Acc*	*ARI*	*Acc*	*ARI*	*Acc*	*ARI*	*Acc*
Zoo	0.42	62.71	0.77	86.14	0.74	81.58	0.66	77.1	0.44	67.33	0.47	69.31	**0.83**	**87.99**
Iris	0.59	70.4	0.76	90.84	0.71	85.64	0.64	78.44	0.72	88.67	0.73	88.33	**0.78**	**91.78**
Ecoli	0.49	62.18	0.6	68.65	0.72	80.38	0.58	70.73	0.53	69.94	**0.73**	**80.85**	0.63	73.55
Segment	0.15	35.76	0.39	61.75	0.35	55.76	0.21	41.17	0.41	63.12	**0.42**	**65.67**	0.41	64.5
Pendigits	0.22	37.05	0.53	71.79	0.54	78.96	0.42	58.9	0.54	76.72	**0.55**	**79.43**	0.54	73.04
BankAuthentication	0.04	60.03	0.09	64.65	0.07	63.59	0.07	62.13	0.05	61.22	0.09	65.16	**0.59**	**88.24**
Balance	0.12	50.41	0.32	63.64	0.13	52.9	0.06	52.11	0.16	53.76	0.4	70.4	**0.53**	**85.41**
Jain	0.26	75.57	0.31	77.75	0.28	76.41	0.35	79.3	0.32	78.55	0.4	80.77	**0.75**	**93.46**
Aggregation	0.74	79.76	0.86	92.7	**0.94**	**96.84**	0.88	92.12	0.73	86.17	0.82	90.12	0.91	95.67
msplice1003	0.53	79.58	0.63	85.82	0.55	80.08	0.17	56.28	0.62	85.51	0.66	86.4	**0.67**	**87.41**
landsat1006	0.28	47.27	0.38	63.92	0.39	64.89	0.29	53.8	0.3	54.5	0.34	60.6	**0.5**	**68.06**

2d-10c-no0	0.8	83.45	0.9	93.79	0.89	92.44	0.89	91.95	0.88	91.76	0.89	93.3	**0.91**	**94.89**
2d-4c-no1	0.79	82.9	0.89	93.33	0.89	92.82	0.86	90.36	0.88	92.67	0.89	92.47	**0.97**	**98.42**
2d-4c-no2	0.85	90.89	0.93	96.92	0.91	96.01	0.9	94.92	0.91	95.77	0.92	95.52	**0.96**	**98.55**
2d-4c-no3	0.9	94.87	0.93	97.64	0.92	97.3	0.92	97.29	0.92	97.33	0.93	97	**0.95**	**98.29**
2d-4c-no4	0.9	93.26	0.98	99.12	**0.99**	**99.61**	0.84	90.75	0.97	98.96	0.97	98	0.98	99.44

Table 4.

Algorithm / **Dataset**	**Cuckoo**		**Semi-Cuckoo**		**NSGAII**		**MOPSO**		**Seeded K-means**		**Constrained K-means**		**ConvexCo**	
	NMI	*F-measure*	*NMI*	*F-measure*	*NMI*	*F-measure*	*NMI*	*F-measure*	*NMI*	*F-measure*	*NMI*	*F-measure*	*NMI*	*F-measure*
Zoo	0.58	0.05	0.77	0.8	0.78	0.67	0.76	0.59	0.67	0.46	0.65	0.59	**0.8**	**0.81**
Iris	0.71	0.29	0.77	0.91	0.77	0.79	0.76	0.74	0.74	0.89	0.75	0.9	**0.8**	**0.92**
Ecoli	0.56	0.11	0.63	0.63	0.69	0.7	0.66	0.49	0.63	0.57	**0.7**	**0.78**	0.62	0.64
Segment	0.27	0.11	0.5	0.63	0.51	0.49	0.36	0.32	0.55	0.68	**0.58**	**0.71**	0.54	0.65
Pendigits	0.51	0.08	0.64	0.72	0.7	0.79	0.57	0.52	0.69	0.77	**0.71**	**0.8**	0.67	0.77
BankAuthentication	0.03	0.49	0.05	0.62	0.08	0.64	0.07	0.59	0.03	0.6	0.06	0.64	**0.61**	**0.91**
Balance	0.13	0.35	0.25	0.59	0.11	0.48	0.07	0.29	0.14	0.51	0.32	0.63	**0.36**	**0.66**
Jain	0.33	0.63	0.36	0.8	0.36	0.8	0.29	0.71	0.37	0.81	0.42	0.83	**0.62**	**0.92**
Aggregation	0.83	0.18	0.88	0.9	**0.95**	**0.96**	0.91	0.88	0.84	0.87	0.87	0.9	0.92	0.95
msplice1003	0.52	0.32	0.57	0.81	0.53	0.82	0.58	0.8	0.59	0.86	0.6	0.87	**0.63**	**0.88**
landsat1006	0.38	0.18	0.5	0.69	0.48	0.63	0.43	0.49	0.43	0.59	0.46	0.64	**0.51**	**0.7**
2d-10c-no0	0.9	0.15	0.93	0.91	0.91	0.92	0.93	0.89	0.93	0.93	0.93	0.9	**0.94**	**0.94**
2d-4c-no1	0.82	0.22	0.85	0.89	0.85	0.9	0.83	0.78	0.83	0.9	0.84	0.36	**0.95**	**0.98**
2d-4c-no2	0.87	0.14	0.92	0.97	0.9	0.95	0.9	0.93	0.9	0.96	0.91	0.39	**0.94**	**0.98**
2d-4c-no3	0.85	0.22	0.91	0.98	0.92	0.98	0.89	0.96	0.91	0.98	0.91	0.39	**0.94**	**0.99**
2d-4c-no4	0.91	0.25	0.96	0.98	**0.97**	**0.99**	0.96	0.98	0.95	0.98	0.95	0.39	**0.97**	**0.99**

Table 4 Continued.

Algorithms / Datasets	Cuckoo	Semi-Cuckoo	NSGAII	MOPSO	Seeded K-means	Constrained K-means	ConvexCo
Zoo	7	2	3	4	6	5	1
Iris	7	2	5	6	4	3	1
Ecoli	7	4	2	5	6	1	3
Segment	7	4	5	6	2.5	1	2.5
Pendigits	7	5	3	6	3	1	3
BankAuthentication	7	2.5	4.5	4.5	6	2.5	1
Balance	6	3	5	7	4	2	1
Jain	7	5	6	3	4	2	1
Aggregation	6	4	1	3	7	5	2
Msplice	6	3	5	7	4	2	1
Landsat	7	3	2	6	5	4	1
2d-10c-no0	7	2	4	4	6	4	1
2d-4c-no1	7	3	3	6	5	3	1
2d-4c-no2	7	2	4.5	6	4.5	3	1
2d-4c-no3	7	2.5	5	5	5	2.5	1
2d-4c-no4	6	2.5	1	7	4.5	4.5	2.5
Average Rank	6.75	**3.09**	**3.69**	5.34	4.78	**2.84**	**1.5**

Table 5.

Datasets / Algorithms	Cuckoo	Semi-Cuckoo	NSGAII	MOPSO	Seeded K-means	Constrained K-means	ConvexCo
Zoo	7	2	3	4	6	5	1
Iris	7	2	5	6	3	4	1
Ecoli	7	6	2	4	5	1	3
Segment	7	4	5	6	3	1	2
Pendigits	7	5	2	6	3	1	4
BankAuthentication	7	3	4	5	6	2	1
Balance	7	3	6	5	4	2	1
Jain	7	5	6	4	3	2	1
Aggregation	7	3	1	4	6	5	2
Msplice	6	4	5	7	3	2	1
Landsat	7	3	2	6	5	4	1
2d-10c-no0	7	2	4	5	6	3	1
2d-4c-no1	2	3	7	6	4	5	1
2d-4c-no2	7	2	3	5	4	6	1
2d-4c-no3	7	2	4	5	3	6	1
2d-4c-no4	6	3	1	7	4	5	2
Average Rank	6.5625	**3.25**	**3.75**	5.3125	4.25	**3.375**	**1.5**

Table 6.

Algorithm / Dataset	Cuckoo	Semi-Cuckoo	NSGAII	MOPSO	Seeded K-means	Constrained K-means	ConvexCo
Zoo	7	3	2	4	5	6	1
Iris	7	2.5	2.5	4	6	5	1
Ecoli	7	4.5	2	3	4.5	1	6
Segment	7	5	4	6	2	1	3
Pendigits	7	5	2	6	3	1	4
BankAuthentication	6.5	5	2	3	6.5	4	1
Balance	5	3	6	7	4	2	1
Jain	6	4.5	4.5	7	3	2	1
Aggregation	7	4	1	3	6	5	2
msplice1003	7	5	6	4	3	2	1
landsat1006	7	2	3	5.5	5.5	4	1
2d-10c-no0	7	3.5	6	3.5	3.5	3.5	1
2d-4c-no1	7	2.5	2.5	5.5	5.5	4	1
2d-4c-no2	7	2	5	5	5	3	1
2d-4c-no3	7	4	2	6	4	4	1
2d-4c-no4	7	3.5	1.5	3.5	5.5	5.5	1.5
Average Rank	6.78	**3.69**	**3.25**	4.75	4.5	**3.31**	**1.72**

Table 7.

Algorithm / Dataset	Cuckoo	Semi-Cuckoo	NSGAII	MOPSO	Seeded K-means	Constrained K-means	ConvexCo
Zoo	7	2	3	4.5	6	4.5	1
Iris	7	2	5	6	4	3	1
Ecoli	7	4	2	6	5	1	3
Segment	7	4	5	6	2	1	3
Pendigits	7	5	2	6	3.5	1	3.5
BankAuthentication	7	4	2.5	6	5	2.5	1
Balance	6	3	5	7	4	2	1
Jain	7	4.5	4.5	6	3	2	1
Aggregation	7	3.5	1	5	6	3.5	2
msplice1003	7	5	4	6	3	2	1
landsat1006	7	2	4	6	5	3	1
2d-10c-no0	7	4	3	6	2	5	1
2d-4c-no1	7	4	2.5	5	2.5	6	1
2d-4c-no2	7	2	4	5	3	6	1

Table 8.

Performance Measures	Paired Algorithms for Comparison	*Paired Differences*					*t-value*	*df*	*p-value*	*Statistical significance*
		MD	*SD*	*Std. Error*	*95% Confidence Interval*					
					Lower	**Upper**				
ARI	(ConvexCo, Constrained K-means	3.56	0.16	0.04	0.07	3.6	2.64	15	0.01	Significant
	(ConvexCo, Semi-Cukoo)	0.1	0.15	0.04	0.06	0.14	2.7		0.02	Significant
	(ConvexCo, NSGAII)	0.12	0.18	0.04	0.07	0.16	2.63		0.02	Significant
Accuracy	(ConvexCo, Semi-Cuckoo)	5.65	7.58	1.89	3.75	7.55	2.98	15	0.009	Significant
	(ConvexCo, Constrained K-means	5.34	8.39	2.1	3.25	7.4	2.55		0.02	Significant
	(ConvexCo, NSGAII)	6.48	10.42	2.61	3.87	9.08	2.48		0.02	Significant
NMI	(ConvexCo, NSGAII)	0.08	0.15	0.03	0.04	0.12	2.17	15	0.04	Significant
	(ConvexCo, Constrained K-means	0.07	0.14	0.04	0.04	0.11	0.02		0.04	Significant
	(ConvexCo, Semi-Cukoo)	0.08	0.14	0.03	0.05	0.12	2.33		0.03	Significant
F-measure	(ConvexCo, NSGAII)	0.073	0.08	0.02	0.05	0.09	3.36	15	0.004	Significant
	(ConvexCo, Constrained K-means	0.18	0.27	0.07	0.12	0.25	2.8		0.01	Significant
	(ConvexCo, Semi-Cukoo)	0.05	0.07	0.01	0.03	0.07	3		0.009	Significant

Table 9.