IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Clustering Mixed Numeric and Categorical Data with Cuckoo Search

JINCHAO JI[1,2,3,4], WEI PANG[5,6], ZAIRONG LI[7], FEI HE[1,2,3,4], GUOZHONG FENG[1,2,3], XIAOWEI ZHAO[1,2,3]

[1]School of Information Science and Technology, Northeast Normal University, Changchun 130117, China
[2]Institute of Computational Biology, Northeast Normal University, Changchun 130117, China
[3]Key Lab of Intelligent Information Processing of Jilin Universities, Northeast Normal University, Changchun 130117, China
[4]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[5]School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, Scotland, EH14 4AS, UK
[6]Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an 710048, China
[7]School of Media Science, Northeast Normal University, Changchun 130117, China

Corresponding author: Wei Pang (pang.wei@abdn.ac.uk), Xiaowei Zhao (zhaoxw303@nenu.edu.cn).

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. (61502093,11501095, 61802057), the Natural Science Foundation of the Education Department of Jilin Province under Grant Nos.(2016504,2016505), the Fundamental Research Funds for the Central Universities under Grant Nos. (2412019FZ048, 2412018JC007, 2412019FZ047, 2412019FZ052), the science and technology research project of "13th Five-Year" of the Education Department of Jilin province under Grant No. JJKH20190290KJ, the China Scholarship Council to Fei He, and the Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an University of Technology, under Contract SKL2017CP01.

ABSTRACT Clustering analysis, as an important technique in data mining, aims to identify the nature groups or clusters of data objects in the attribute space. Data objects in real-world applications are commonly described by both numeric and categorical attributes. In this research, considering that the partitional clustering algorithms designed for this type of mixed data are prone to get trapped into local optima and the cuckoo search approach is efficient in solving global optimization problems, we propose CCS-K-Prototypes, a novel partitional Clustering algorithm based on Cuckoo Search and K-Prototypes, for clustering mixed numeric and categorical data. To deal with different types of attributes, we develop a novel representation for candidate solutions, and suggest two formulas for the cuckoo to search for the potential solution around the existing solutions or in the entire attribute space. Finally, the performance of the proposed algorithm is assessed by a series of experiments on five benchmark datasets.

INDEX TERMS Data Clustering, cuckoo search, mixed data, numeric and categorical attributes

I. INTRODUCTION

Clustering analysis aims to detect the nature groups or clusters of data objects in attributes space, and it is one of the most important techniques in data mining [1], [2]. Clustering algorithms are used in a wide range of fields such as social media analysis [3], information retrieval [4], [5], image analysis [6], privacy preserving [7], text analysis [8], and bioinformatics [9], [10]. The goal of clustering is to group data objects into clusters such that the data objects in the same cluster are as similar as possible and the ones from different clusters are as dissimilar as possible [11]. Clustering algorithms can be considered falling into two types: hierarchical and partitional [2]. In hierarchical clustering algorithms, data objects are distributed into a dendrogram of the nested partitions according to a divisive or agglomerative strategy [12]. Whereas in partitional clustering algorithms, data objects are divided into a given number of clusters by minimizing an objective cost function.

The k-means algorithm is a simple and popular centre-based partitional clustering algorithm [13]. Considering the uncertainty of data objects, Bezdek, Ehrlich, and Full introduced the fuzzy k-means algorithm [14]. The k-means algorithm and its fuzzy version are originally designed for the datasets with numeric attributes. However, in many real-world applications, data objects are described by both numeric and categorical attributes. To deal with this type of data, the k-prototypes algorithm was introduced by Huang [15]. Considering the fuzzy nature of the data objects amongst clusters, the fuzzy k-prototypes algorithm was introduced by Bezdek *et al* [16]. In addition, several extensions of the k-prototypes algorithms were proposed by taking the significance of attribute and the representation of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

cluster's centre into account [11], [17], [18], [19]. Lam, Wei, and Wunsch proposed the approach UFLA, which is based on the unsupervised feature learning (UFL) and fuzzy adaptive resonance theory (ART) [20]. Based on the density clustering, Chen and He introduced the algorithm ACC-FSFDP, which is a self-adaptive peak density clustering algorithm [21].

However, one issue associated with many existing clustering algorithms, including k-prototypes algorithms, is that they are prone to get trapped into local optima. Over the past decades, meta-heuristic algorithms have been widely used to perform global search on complex search space of many problems. Holland *et al* proposed the well-known genetic algorithm, which models the process of the biological evolution on the basis of Charles Darwin's theory of natural selection [22]. Kennedy and Eberhart developed the particle swarm optimization (PSO) on the basis of the swarm behaviour such as fish and bird schooling [22]. Lucic and Teodorović presented a bee colony optimization approach, which is inspired by the foraging behaviour of bee swarm in the real world [23].

Inspired by the brood parasitic behaviour of some cuckoo species, Yang and Deb proposed the cuckoo search (CS) algorithm [24]-[26]. The CS is enhanced by the Lévy flight behaviour of some birds and fruit flies, and has good global convergence property [26]. Yang and Deb also developed the multiobjective version of the cuckoo search [27]. Marichelvam, Prabaharan, and Yang developed an improved cuckoo search approach for flow shop scheduling problem [28]. Adnan, Razzaque *et al* introduced the cuckoo search to deal with the cluster arrangement of the wireless sensor network [29]. Goel, Sharma, and Bedi proposed a cuckoo search clustering algorithm, which used Davies-Bouldin index as fitness function [30]. In the comparative study, Senthilnath, Das, and Omkar *et al* pointed out that the cuckoo search was efficient on clustering problems [31]. Based on the cuckoo search strategy and k-modes algorithm, Lakshmi, Visalakshi, and Shanthi *et al* developed the algorithm Cuckoo-K-Modes for dealing with the categorical data [32].

In real-world applications, the collected data with both numeric and categorical attributes are ubiquitous. Based on the cuckoo search strategy and k-prototypes algorithm, Lakshmi, Visalakshi, and Shanthi introduced the cuckoo search based k-prototype algorithm [33]. In Lakshmi *et al.*'s algorithm, the cluster centres are initialized by the cuckoo search and then updated by the k-prototypes algorithm. In this research we aim to develop a novel CS-based clustering algorithm for the mixed data with both numeric and categorical attributes. We first introduce a novel representation for candidate solutions, and then integrate this representation with the cuckoo search framework to cluster mixed data. Finally, we analyse the time and space complexity of the proposed approach, and test it on selected datasets. The differences between our proposed algorithm and the Lakshmi *et al.*'s algorithm are summarized as follows: firstly, we give the representation for candidate solutions; secondly, we give the calculation method for the fitness value of a candidate solution; thirdly, we give the calculation methods of the candidate solution for local and global search, respectively; fourthly, the process of the CCS-K-Prototypes algorithm is different from the Lakshmi *et al.*'s algorithm.

The rest of this paper is organized as follows: we first briefly review some related work in Section II. Then, we present the proposed approach in Section III, and report the experimental results which demonstrate the advantages of the proposed method in Section IV. Finally, we conclude this paper and explore the future directions in Section V.

II. MOTIVATION AND RELATED WORK

In this section, we first present the notations used throughout this paper, the clustering task, the k-prototypes algorithm, and then depict the idea of cuckoo search strategy.

A. NOTATIONS

Let $X = \{X_1, X_2, ..., X_n\}$ be a dataset consisting of n data objects and X_i $(1 \le i \le n)$ be a data object with m attributes $A_1, A_2, ..., A_m$. Then all possible values of an attribute A_j form the domain of values indicated by $Dom(A_j)$. The domain of values associated with mixed data has two types: numeric and categorical. The numeric domain is represented by continuous real numbers. Whereas the categorical domain is represented by a finite set without any natural ordering (such as gender, colour), which is usually denoted by $Dom(A_j) = \{a_j^1, a_j^2, ..., a_j^t\}$, where t is the number of category values of the categorical attribute A_j in the dataset X. A data object X_i is logically represented as a conjunction of attribute-value pairs $[A_1 = x_{i1}] \wedge [A_2 = x_{i2}] \wedge ... \wedge [A_j = x_{ij}] \wedge ... \wedge [A_m = x_{im}]$, where $x_{ij} \in Dom(A_j)$ for $1 \le j \le m$. For ease of description, we represent X_i as a vector $[x_{i1}, x_{i2}, ..., x_{im}]$. We assume that every data object has exactly m attributes for the datasets considered in this paper.

B. THE CLUSTERING TASK

Clustering is the process of identifying the nature groups or clusters of data objects [2]. Let $X = \{x_1, x_2, ..., x_n\}$ denote a dataset with n data objects and x_i be a data object described by m attributes. The aim of clustering is to determine a partition $P = \{C_1, C_2, ..., C_k\}$ which is subject to the following constraints: $\forall j : C_j \neq \phi$, $\forall i \neq j$: $C_i \bigcap C_j = \phi$, and $\sum_{i=1}^{k} C_i = X$. After partition, the data objects in the same cluster are as similar as possible whereas the ones from different clusters are as dissimilar as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

possible. For achieving good clustering results, one popular way is to minimize the following objective function:

$$E(U,z)=\sum_{i=1}^{n}\sum_{j=1}^{k}u_{ij}d(x_i,z_j)\,, \tag{1}$$

where z is the set of cluster centres and z_j is the centre of Cluster j, x_i is the data object i, and $d(x_i,z_j)$ is the distance between the data object x_i and cluster centre z_j, and U is the partition matrix: if a data object i belongs to Cluster j, $u_{ij}=1$; otherwise $u_{ij}=0$. Given all cluster centres z, u_{ij} is calculated as follows:

$$u_{ij}=\begin{cases}1 & \text{if } j=\text{argmin}_{t\in\{1,2,\ldots,k\}}d(\mathrm{x}_i,\mathrm{z}_t),\\ 0 & \text{otherwise,}\end{cases} \tag{2}$$

where $i\in\{1,2,\ldots,n\}$ and $j\in\{1,2,\ldots,k\}$.

C. THE K-PROTOTYPES ALGORITHM

The k-prototypes algorithm was first introduced by Huang for clustering the data with both numeric and categorical attributes [15]. This algorithm aims to partition the dataset X into k clusters by minimizing the following cost function:

$$E(U,\mathrm{Z})=\sum_{l=1}^{k}\sum_{i=1}^{n}u_{il}d(x_i,\mathrm{Z}_l)\,, \tag{3}$$

where Z_l is the prototype of the cluster l; u_{il} $(0\le u_{il}\le 1)$ is an element of the partition matrix $U_{n\times k}$; and $d(x_i,\mathrm{Z}_l)$ is the dissimilarity measure which is given as follows:

$$d(x_i,\mathrm{Z}_l)=\sum_{j=1}^{m}d(x_{ij},\mathrm{z}_{lj})\,. \tag{4}$$

In (4), $d(x_{ij},\mathrm{z}_{lj})$ is formulated as:

$$d(x_{ij},\mathrm{z}_{lj})=\begin{cases}(x_{ij}-z_{lj})^2 & \text{if the } l\text{th attribute is numeric,}\\ \mu_l\delta(x_{ij},\mathrm{z}_{lj}) & \text{if the } l\text{th attribute is categorical,}\end{cases} \tag{5}$$

where $\delta(p,q)=0$ if the values of p and q are the same; $\delta(p,q)=1$ if the values of p and q are different; μ_l is a weight for categorical attributes in the cluster l. If the jth attribute is the numeric attribute, z_{lj} is the mean of the jth numeric attribute in the cluster l; if the jth attribute is the categorical one, z_{lj} is the mode of the jth categorical attribute in the cluster l. The procedure of the k-prototypes algorithm is depicted as follows:

Step 1. Randomly select k data objects from the dataset X as the initial prototypes of clusters.

Step 2. Allocate each data object in the dataset X to the cluster with its nearest prototype according to (4). Update the prototype of cluster after each allocation.

Step 3. Once all data objects have been allocated, re-evaluate the similarity of data objects against the current prototypes. If it is found that a data object's nearest prototype locates in another cluster rather than the current one, reallocate this data object to that cluster and update the prototypes for both clusters.

Step 4. If no data objects have changed clusters after a full circle test of X, terminate the algorithm; otherwise, go to Step 3.

D. THE CUCKOO SEARCH STRATEGY

Cuckoos are the interesting birds with charming sounds and aggressive breed behaviour. Many of them lay their eggs into the nest of the other host birds. If a host bird finds the alien eggs, it will either cast these eggs outside the nest or quit its nest and builds a new one somewhere else. The eggs of cuckoo generally hatch slightly earlier than the eggs of the host birds. The cuckoo chick will push the host eggs out of the nest after it hatched (see [24] for more details). Inspired by these interesting breeding behaviour, Yang and Deb introduced the cuckoo search strategy [24]. This strategy has three idealized rules: 1) the cuckoo lays one egg at a time, and puts its egg into a randomly selected nest; 2) the nests, which contain high quality eggs, will pass to the next generations; 3) the number of obtainable host nests is fixed, and the cuckoo's egg is perceived by the host bird with a probability $pro_a\in[0,\,1]$. Given the objective function $f(x)$, $x=(x_1,\ldots,x_d)^T$, the process of cuckoo search is briefly given as follows:

Initialization: Initialize the population of n host nests x_i $(i=1,2,\ldots,n)$

While (the stop criterion is not met)

Get a cuckoo x_i in a random way by Lévy flights, and calculate its quality or fitness F_i;

Pick up a nest x_j from the population randomly, and evaluate its quality or fitness F_j;

If ($F_i>F_j$)

Replace the nest x_j with the new nest x_i;

End if

Abandon a fraction (pro_a) of the worse nests, and generate new ones by Lévy flights;

Keep the nests with the highest quality;

Rank the nests and keep the nest with the highest quality or fitness;

End while

Output the best solution;

In the process of cuckoo search, the stop criterion will be either the number of max generations or other termination criteria such as the best clustering solution does not improve for a given number of generations. The Lévy flight is a random walk in essence, and its step length obeys the Lévy distribution.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

III. THE PROPOSED METHOD

In this section, we first describe our proposed CCS-K-Prototypes (Clustering based on Cuckoo Search and K-Prototypes) clustering approach, and we then give an example of the CCS-K-Prototypes algorithm. Finally, we analyse the time and space complexity of the CCS-K-Prototypes algorithm.

A. THE PROPOSED APPROACH

In this section, we proposed a novel clustering algorithm on the basis of the cuckoo search strategy and the k-prototypes algorithm. In the search process of cuckoo, the egg in a nest represents a candidate solution. As aforementioned, one nest only contains one egg, and each cuckoo lays one egg at a time. Thus, a cuckoo and a nest correspond to a candidate solution. In our algorithm, the nest, cuckoo, and the egg have the same meaning. In centre-based clustering algorithms, the clustering results depend on the cluster centres: the clustering results are determined once the cluster centres are obtained. Therefore, the clustering task can be seen as the process of searching for good cluster centres, and the set of cluster centres represents a candidate solution. Suppose the dataset with n data objects has k clusters, the set of cluster centres $z_i = \{z_{i1}, z_{i2}, ..., z_{ik}\}$ represents a candidate solution. For the data with both numeric and categorical attributes, the prototype, which is the combination of the mean and mode, is used to represent the cluster's centre. In such clustering tasks, the fitness value of a candidate solution $z_i = \{z_{i1}, z_{i2}, ..., z_{ik}\}$ is given as follows:

$$F(z_i) = \sum_{e=1}^{n}\sum_{j=1}^{k} u_{ej} d(x_e, z_{ij}), \tag{6}$$

where u_{ej} is the element of the partition matrix U which is determined according to the cluster centres $z_i = \{z_{i1}, z_{i2}, ..., z_{ik}\}$. Given the cluster centre z_i, u_{ej} is calculated as follows:

$$u_{ej} = \begin{cases} 1 & \text{if } j = \text{argmin}_{t\in\{1,2,...,k\}} d(x_e, z_{it}), \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where $e \in \{1, 2, ..., n\}$, and $j \in \{1, 2, ..., k\}$. In (6), the dissimilarity measure $d(x_e, z_{ij})$ is formulated as:

$$d(x_e, z_{ij}) = \sum_{s=1}^{m} \varphi(x_{es}, z_{ijs}), \tag{8}$$

where $\varphi(x_{es}, z_{ijs})$ is given by:

$$\varphi(x_{es}, z_{ijs}) = \begin{cases} \left(\dfrac{x_{es} - z_{ijs}}{\max_s - \min_s}\right)^2 & \text{if the } s\text{th attribute is numeric,} \\ \mu_s \delta(x_{es}, z_{ijs}) & \text{if the } s\text{th attribute is categorical,} \end{cases} \tag{9}$$

where x_{es} is the value of the sth attribute of a data object e, μ_s is the weight of the sth categorical attribute in the cluster j, and $\max_s / \min_s$ is the maximum/minimum value of the sth attribute in the dataset X. If the sth attribute is a numeric attribute, z_{ijs} is the mean of the sth numeric attribute in the cluster j; if the sth attribute is a categorical one, z_{ijs} is the most frequent value, i.e., the mode, of the sth categorical attribute in the cluster j.

For a cuckoo i, the new solution z_i^{t+1} is generated from its current solution z_i^t and the best solution z_{best}^t, which is given as follows:

$$z_i^{t+1} = getNextSolution(z_i^t, z_{best}^t). \tag{10}$$

Let z_{ijs}^{t+1}, $z_{best\_js}^t$ be the value of the sth attribute of the jth cluster centre of the candidate solution z_i^{t+1}, and z_{best}^t, respectively. If the sth attribute is a categorical attribute, z_{ijs}^{t+1} is the categorical value, which is selected randomly from the set of values Vals= $\{z_{ijs}^t, z_{best\_js}^t\}$. If the sth attribute is a numeric one, z_{ijs}^{t+1} is calculated by:

$$z_{ijs}^{t+1} = z_{ijs}^t + \alpha \times sl \times |z_{best\_js}^t - z_{ijs}^t|, \tag{11}$$

where α is the coefficient which is related to the scale of the problem to be solved, and the symbol $|.|$ denotes the absolute value. In (11), sl is the step length of Lévy flight. In other words, sl obeys Lévy distribution. As pointed out by Yang [22], the Mantega's algorithm can be used to generate the step length of Lévy flight. In the Mantega's algorithm, the step length sl is calculated by:

$$sl = \frac{u}{|v|^{1/\beta}}, \tag{12}$$

where u and v obey the normal distribution as follows:

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2). \tag{13}$$

In (13),

$$\sigma_u = \left\{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\beta\,\Gamma((1+\beta)/2)2^{(\beta-1)/2}}\right\}^{1/\beta}, \quad \sigma_v = 1, \tag{14}$$

where $\Gamma(\cdot)$ is the gamma function as follows:

$$\Gamma(1+\beta) = \int_0^\infty t^\beta e^{-t} dt. \tag{15}$$

In the proposed approach, the worst nest will be abandoned by a cuckoo with the probability pro_a. For the abandoned nest z_{aban}^t, the new nest z_{new}^t is generated as follows:

$$z_{new}^t = getNewSolution(z_{aban}^t, z_f^t, z_g^t), \tag{16}$$

where z_{aban}^t denotes the abandoned nest, and the two solutions z_f^t and z_g^t are selected randomly from the set of candidate solutions. Let $z_{new\_js}^t$, $z_{aban\_js}^t$, z_{fjs}^t, and z_{gjs}^t be the values of the sth attribute of the jth cluster centre in the solution z_{new}^t, z_{aban}^t, z_f^t, and z_g^t, respectively. If the sth attribute is a categorical one, $z_{new\_js}^t$ is the categorical value which is selected in a random way from the collection of values Vals= $\{z_{aban\_js}^t, z_{fjs}^t, z_{gjs}^t\}$. If the sth attribute is the numeric one, $z_{new\_js}^t$ is calculated by:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

$$z^{t}_{new\_js} = z^{t}_{aban\_js} + \varepsilon \times | z_{fjs} - z_{gjs} |, \quad (17)$$

where ε is a random number between 0 and 1. Based on the above descriptions, we give the process of the proposed approach in Algorithm CCS-K-Prototypes.

Algorithm CCS-K-Prototypes

Input: The cluster number k, the maximum number of generations (maxGen), the abandon probability pro_a, and the number of nests N, beta;

Output: The best solution and the clustering result.

1: **Initialization:** Initialize each nest z_i $(i = 1, 2, ..., N)$ in the population P randomly. More specifically, the candidate solution $z_i (1 \le i \le N)$ is initialized as the k data objects, which are randomly selected from the dataset X; set the generation number $t = 0$ for these candidate solutions.
2: Evaluate and pick up the best nest z_{best} from the population P;
3: **While (*t*<maxGen)**
4: Get a cuckoo z_i according to (10), and calculate its fitness $F(z_i)$ according to (6);
5: **If (** $F(z_i) < F(z_{best})$ **)**
6: Replace the best nest z_{best} by the new solution z_i;
7: **End if**
8: Assign a random value between the range [0, 1] to the probability pro_t;
9: **If (** $pro_t < pro_a$ **)**
10: Abandon the worst nest z_{worst}, and build a new one z_{new} according to (16);
11: Calculate the fitness for the new nest z_{new} according to (6);
12: **End if**
13: **If (** $F(z_{new}) < F(z_{worst})$ **)**
14: Replace the worse nest z_{worst} by the new one z_{new};
15: **End if**
16: Rank the solutions and keep the current best solution z_{best};
17: $t=t+1$;
18: **End while**
19: Output the best solution z_{best} and output the final clustering result

B. AN EXAMPLE OF THE PROPOSED ALGORITHM

In this section, for better illustrating our algorithm, we use a simple synthetic dataset to demonstrate the work process of the proposed CCS-K-Prototypes algorithm. The synthetic dataset contains seven data objects, each of which is described by two numeric attributes (age and height) and two categorical attributes (gender and hobby). These data objects are listed in Table 1.

TABLE 1. The data objects of the synthetic dataset

Attributes / ID	Gender	Age	Height(cm)	hobby
1	male	18	175	writing
2	female	24	165	music
3	female	23	175	tennis
4	male	45	185	football
5	male	35	195	basketball
6	male	26	180	tennis
7	female	22	172	music

Suppose this dataset has two clusters. Let the cluster number k is 2; the *maxGen* is 3; the abandon probability pro_a is 0.3; the number of nests N is 3; beta is 1.5. In the initialization stage, the three nests are initialized as three sets of two randomly chosen data objects. Suppose nest z_1 is the set of data objects 5 and 6, nest z_2 is the set of data objects 1 and 7, and nest z_3 is the set of data objects 3 and 6. We list the nest z_1, z_2, and z_3 as follows:

z_1={male, 35.0, 195.0, basketball; male, 26.0, 180.0, tennis }
z_2={male, 18.0, 175.0, writing; female, 22.0, 172.0, music }
z_3={female, 23.0, 175.0, tennis; male, 26.0, 180.0, tennis }

The fitness values of nests z_1, z_2, and z_3, which are calculated using (6), are 7.75, 6.14, and 6.12, respectively. The nest z_3 with the minimum value of fitness is chosen as the best nest z_{best}.

In the local search of a cuckoo, nest z_2 is randomly chosen as the current nest z_i. According to (10), z_i is updated as:

z_i={male, 18.0, 175.03, tennis; female, 21.99, 171.96, music}

The fitness value of nest z_i is 6.14, which is higher than that of the best nest z_{best}. Therefore, the best nest keeps unchanged.

In the global search of a cuckoo, the pro_t is randomly set as 0.23, which is lower than the abandon probability pro_a. The worst nest z_1 with the highest fitness value 7.75 is abandoned, and a new one is generated according to (16). The new nest is listed as follows:

z_{new}={male, 18.0, 175.03, tennis; female, 21.99, 171.96, music}

The fitness value of the new nest z_{new} is 7.70, which is lower than that of the worst nest. Therefore, the worst nest is replaced by the new one. The updated candidate nests are listed as follows:

z_1={male, 18.0, 175.03, tennis; female, 21.99, 171.96, music}
z_2={male, 18.0, 175.0, writing; female, 22.0, 172.0, music}
z_3={female, 23.0, 175.0, tennis; male, 26.0, 180.0, tennis}

The fitness values of z_1, z_2, and z_3 are 7.70, 6.14, and 6.12, respectively. The best nest is updated as nest z_3.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

This search process will repeat until the number of generations reaches the maximum number of generations (maxGen). When the search process terminates, the best solution is:

z_{best}={female, 23.0, 175.0, tennis; male, 26.0, 180.0, tennis}

The clustering result is obtained according to the best solution z_{best}. There are two clusters in the clustering result. Let the symbol Cluster1 denotes the first cluster, and the symbol Cluster2 indicates the second cluster. In each cluster, the ID of data objects is listed as following:

Cluster1={2,3,7}

Cluster2={1,4,5,6}.

C. ALGORITHM COMPLEXITY ANALYSIS

In this section, we analyse the time and space complexity of the proposed CCS-K-prototypes approach. The time complexity of the proposed method mainly contains three parts: the initialization of host nests, the search for candidate solutions, and the calculation of the fitness of candidate solutions. The computational cost of these three parts are $O(Nk)$, $O(km)$, and $O(nkm)$, respectively. Here N is the number of host nests, n is the number of data objects in the dataset X, m is the number of attributes, and k is the number of clusters. Therefore, the overall time complexity of the proposed approach is $O\left(Nk+s\left(km+nkm\right)\right)$, where s is the number of generations. For space complexity, it requires $O(mn)$ to store the dataset X, $O(Nkm)$ to store the candidate solutions, and $O(nk)$ to store the partition matrix. Therefore, the overall space complexity of our method is $O(mn+Nkm+nk)$.

IV. EXPERIMENTS AND DISCUSSION

To assess the performance of CSS-K-Prototypes, we execute the CCS-K-Prototypes on five datasets: zoo, heart disease, credit approval, soybean, and breast cancer, which are obtained from the well-known UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets.html). The details of these datasets are given in Table 2.

TABLE 2. The details of the datasets

Dataset	Number of numeric attributes	Number of categorical attributes	Number of data objects	Number of classes
Zoo	1	16	101	7
Heart disease (Case 1)	6	9	303	5
Heart disease (Case 2)	6	8	303	2
Credit approval	6	10	690	2
Soybean	0	36	47	4
Breast cancer	9	2	699	2

In this work, we adopt Yang's accuracy measures [34] and the Rand Index [35], which are two commonly used criteria, to evaluate the obtained clustering results. In Yang's method, the accuracy (AC), precision (PR), and recall (RE) is formulated as follows:

$$AC=\frac{\sum_{i=1}^{k}a_i}{n}, \tag{18}$$

$$PR=\frac{\sum_{i=1}^{k}\frac{a_i}{a_i+b_i}}{k}, \tag{19}$$

$$RE=\frac{\sum_{i=1}^{k}\frac{a_i}{a_i+c_i}}{k}, \tag{20}$$

where a_i is the number of data objects that are correctly assigned to the class C_i, b_i is the number of data objects that are incorrectly assigned to the class C_i, c_i is the number of data objects that are incorrectly denied from the class C_i, n is the number of data objects in a dataset, and k is the number of classes contained in the dataset. In Yang's measures, AC has the same meaning as the clustering accuracy r given in [36], and PR has the same meaning as the purity of clusters given in [21]. Given a dataset $X=\{x_1,x_2,...,x_n\}$ as well as two partitions of this dataset: $Y=\{\mathrm{y}_1,\mathrm{y}_2,...,\mathrm{y}_{h_1}\}$ and $Y'=\{\mathrm{y}_1',\mathrm{y}_2',...,\mathrm{y}_{h_2}'\}$, the Rand Index (RI) [35] is given as follows:

$$RI=\frac{\sum_{i=1,j=2;i<j}^{n}\alpha_{ij}}{\binom{n}{2}}, \tag{21}$$

where α_{ij} is determined as follows:

$$\alpha_{ij}=\begin{cases}1, & \text{if there exist } h \text{ and } h' \text{ such that both } x_i \\ & \text{and } x_j \text{ are in both } \mathrm{y}_h \text{ and } y_{h'}', \\ 1, & \text{if there exist } h \text{ and } h' \text{ such that } x_i \text{ is in both} \\ & \mathrm{y}_h \text{ and } y_{h'}' \text{ while } x_j \text{ is in neither } \mathrm{y}_h \text{ nor } y_{h'}', \\ 0, & \text{otherwise.}\end{cases} \tag{22}$$

According to these measures, the higher values of AC, PR, RE, and RI denote the better clustering result. In the performance analysis, we run our proposed CCS-K-Prototypes algorithm, the K-prototypes algorithm [15], the SBAC algorithm [37], the KL-FCM-GM algorithm [19], the EKP algorithm [38], and the ABC-K-Prototypes algorithm [39] to cluster five different datasets. For each dataset, we run twenty trials, and the average values of AC, PR, RE, and RI are calculated. The clustering results of the ACC-FSFDP algorithm reported in [21] are also supplied for comparison. Then we compare the clustering result of the proposed CCS-K-Prototypes algorithm with those of the other six popular

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

algorithms according to the average of AC, PR, RE, and RI, respectively. All algorithms except ACC-FSFDP are implemented in Java language and run on a PC with Intel (R) Core (TM) i7, 3.4GHz CPU, and 16GB RAM. In all algorithms except ACC-FSFDP, the number of clusters k is set as the number of classes provided by the class information of the datasets. We point out that other class information was not used in the clustering process except the number of classes. The other parameters of the K-prototypes algorithm, the SBAC algorithm, and the KL-FCM-GM algorithm, the EKP algorithm, and the ABC-K-Prototypes algorithm are set the same as those given in their original papers. In the proposed CCS-K-Prototypes algorithm, we set the maximum generations as 2,500 by the rule of thumb, and use the number of nests $N = 15,\ 20,\ 25,\ 30,\ 35,\ 40$, the abandon probability $pro_a = 0.15,\ 0.2,\ 0.25,\ 0.3$, and beta=1.5, which are the typical values given in the original cuckoo search algorithm [22], [24], [25].

We begin our experiments on the zoo dataset. This dataset contains 101 data objects, each of which is characterized by one numeric attribute and 16 categorical attributes. The last categorical attribute is the class attribute, and has seven values. The data objects in the zoo dataset therefore belong to one of the seven classes. Figs. 1a-1d give the effect of the number of nests N and the abandon probability pro_a (*pa* for short) on the values of AC, PR, RE, and RI of the proposed CCS-K-Prototypes algorithm for clustering this dataset, respectively. From these figures, we can see that AC achieves its highest value when the number of nests N equals to 40, and the abandon probability *pa* equals to 0.2; PR obtains its highest value when the number of nests N equals to 30, and the abandon probability *pa* equals to 0.2; RE achieves its highest value when the number of nests N equals to 30, and the abandon probability *pa* equals to 0.3; RI obtains its highest value when the number of nests N equals to 20, and the abandon probability *pa* equals to 0.25. The results in Figs. 1a-1d illustrate that the performance of the proposed CCS-K-Prototypes algorithm on the zoo dataset is affected by the number of nests N and the abandon probability *pa*, and the CCS-K-Prototypes algorithm can achieve reasonably good results within the given range of N and *pa*. In Tables 3a-3d, we summarize the clustering results of the CCS-K-Prototypes, the K-Prototypes, the SBAC, the KL-FCM-GM, the EKP, and the ABC-K-Prototypes algorithms on the zoo dataset according to AC, PR, RE, and RI, respectively. In Tables 3a-3b, we also list the clustering results of the algorithm ACC-FSFDP, which are taken from [21]. The results in Tables 3a-3d indicate that our proposed CCS-K-Prototypes approach achieves the highest values on AC, PR, RE, and obtains a near highest value on RI.

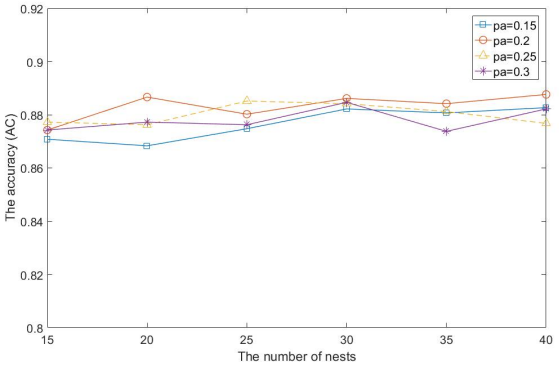

FIGURE 1a. The accuracy (AC) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the zoo dataset

TABLE 3a. The accuracy (AC) of the seven algorithms on the zoo dataset

Algorithms	AC
K-Prototypes	0.806
SBAC	0.426
KL-FCM-GM	0.870 ($\alpha = 1.3$)
EKP	0.628
ABC-K-Prototypes	0.886
ACC-FSFDP	0.874
CCS-K-Prototypes	**0.888** (N=40, pa=0.2)

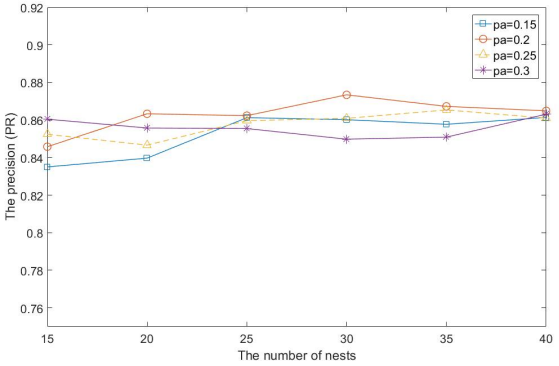

FIGURE 1b. The precision (PR) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the zoo dataset

TABLE 3b. The precision (PR) of the seven algorithms on the zoo dataset

Algorithms	PR
K-Prototypes	0.827
SBAC	0.484
KL-FCM-GM	0.844 ($\alpha = 1.3$)
EKP	0.729
ABC-K-Prototypes	0.861
ACC-FSFDP	0.862
CCS-K-Prototypes	**0.873** (N=30, pa=0.2)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

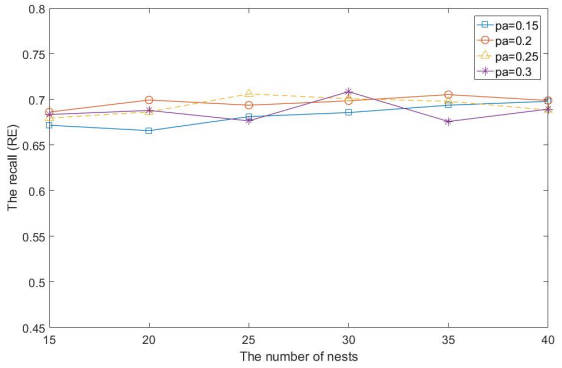

FIGURE 1c. The recall (RE) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the zoo dataset

TABLE 3c. The recall (RE) of the six algorithms on the zoo dataset

Algorithms	RE
K-Prototypes	0.636
SBAC	0.172
KL-FCM-GM	0.685 ($\alpha = 1.3$)
EKP	0.419
ABC-K-Prototypes	**0.718**
CCS-K-Prototypes	0.709 (*N*=30, *pa*=0.3)

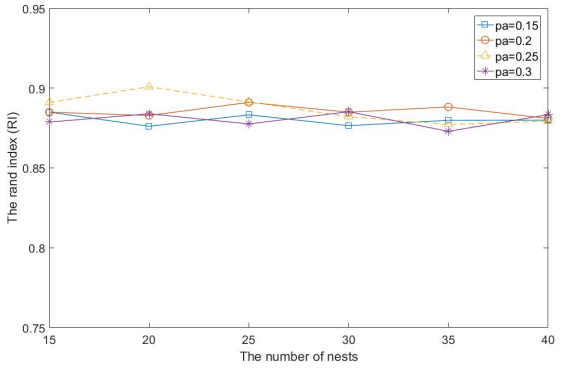

FIGURE 1d. The rand index (RI) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the zoo dataset

TABLE 3d. The rand index (RI) of the six algorithms on the zoo dataset

Algorithms	RI
K-Prototypes	0.857
SBAC	0.648
KL-FCM-GM	**0.918** ($\alpha = 1.8$)
EKP	0.601
ABC-K-Prototypes	0.894
CCS-K-Prototypes	0.901 (*N*=20, *pa*=0.25)

The heart disease dataset has 303 patient instances, each of which is described by six numeric attributes and nine categorical attributes. The last two attributes are class attributes. When we take the 15*th* attribute as its class attribute, the data objects in this dataset belong to one of five classes (s1, s2, s3, s4, and H), and each of them is described by 14 attributes; when we take the 14*th* attribute as its class attribute, the data objects in this dataset belong to one of two classes (buff, sick), and each of them is described by 13 attributes. For the first case, Figs. 2a-2d present the effect of the number of nests *N* and the abandon probability pro_a (*pa* for short) on the values of AC, PR, RE, and RI of the proposed CCS-K-Prototypes algorithm for clustering this dataset, respectively. From these figures, we can see that AC achieves its highest value when the number of nests *N* equals to 30, and the abandon probability *pa* equals to 0.2; PR obtains its highest value when the number of nests *N* equals to 40, and the abandon probability *pa* equals to 0.15; RE achieves its highest value when the number of nests *N* equals to 35, and the abandon probability *pa* equals to 0.15; RI obtains its highest value when the number of nests *N* equals to 35, the abandon probability *pa* equals to 0.25. The results in Figs. 2a-2d show that the performance of the proposed CCS-K-Prototypes algorithm on the heart disease dataset (first case) is affected by the number of nests *N* and the abandon probability *pa*, and the CCS-K-Prototypes algorithm can achieve reasonably good results within the given range of *N* and *pa*. In Tables 4a-4d, we list the clustering results of the CCS-K-Prototypes, the K-Prototypes, the SBAC, the KL-FCM-GM, the EKP, and the ABC-K-Prototypes algorithms on the heart diseases dataset (first case) according to AC, PR, RE, and RI, respectively. The results in Tables 4a-4d indicate that our proposed CCS-K-Prototypes approach achieves the highest values or near highest values on most of the four measures.

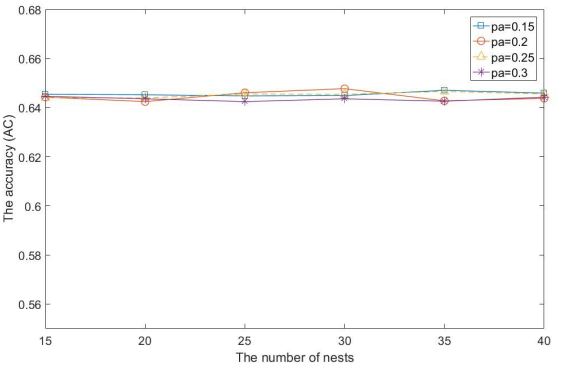

FIGURE 2a. The accuracy (AC) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the heart disease dataset (5 classes and 14 attributes)

TABLE 4a. The accuracy (AC) of the six algorithms on the heart disease dataset (5 classes and 14 attributes)

Algorithms	AC
K-Prototypes	0.547
SBAC	0.545
KL-FCM-GM	**0.653** ($\alpha = 1.2$)
EKP	0.545
ABC-K-Prototypes	0.648
CCS-K-Prototypes	0.648 (*N*=30, *pa*=0.2)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

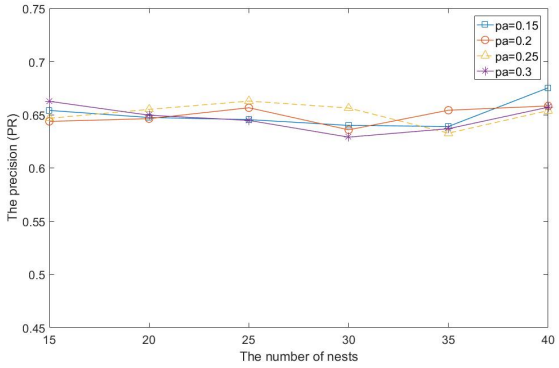

FIGURE 2b. **The precision (PR) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the heart disease dataset (5 classes and 14 attributes)**

TABLE 4b. **The precision (*PR*) of the six algorithms on the heart disease dataset (5 classes and 14 attributes)**

Algorithms	PR
K-Prototypes	0.521
SBAC	0.566
KL-FCM-GM	**0.766** ($\alpha = 1.9$)
EKP	0.109
ABC-K-Prototypes	0.658
CCS-K-Prototypes	0.675 (*N*=40, *pa*=0.15)

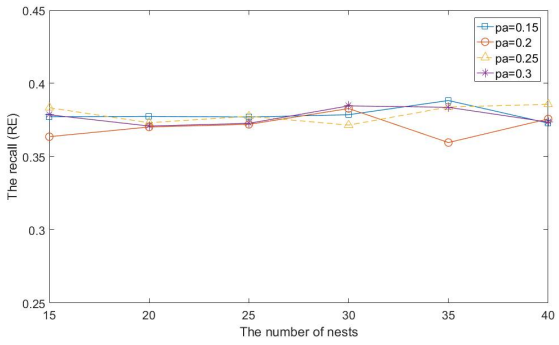

FIGURE 2c. **The recall (RE) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the heart disease dataset (5 classes and 14 attributes)**

TABLE 4c. **The recall (*RE*) of the six algorithms on the heart disease dataset (5 classes and 14 attributes)**

Algorithms	RE
K-Prototypes	0.216
SBAC	0.2
KL-FCM-GM	**0.395** ($\alpha = 1.4$)
EKP	0.2
ABC-K-Prototypes	0.379
CCS-K-Prototypes	0.388 (*N*=35, *pa*=0.15)

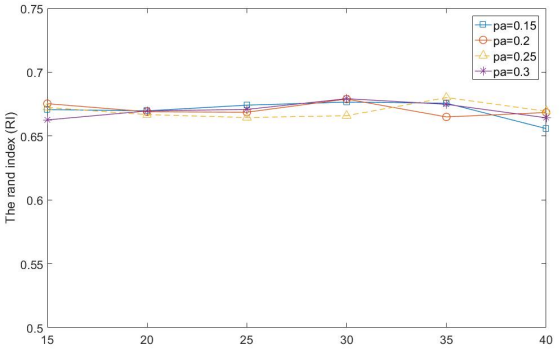

FIGURE 2d. **The rand index (RI) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the heart disease dataset (5 classes and 14 attributes)**

TABLE 4d. **The rand index (RI) of the six algorithms on the heart disease dataset (5 classes and 14 attributes)**

Algorithms	RI
K-Prototypes	0.601
SBAC	0.503
KL-FCM-GM	0.673 ($\alpha = 1.2$)
EKP	0.355
ABC-K-Prototypes	0.667
CCS-K-Prototypes	**0.680** (*N*=35, *pa*=0.25)

For the second case where each data object in the heart disease dataset is characterized by 13 attributes and the 14th attribute is taken as its class attribute. Figs. 3a-3d illustrate the effect of the number of nests N, and the abandon probability pro_a (*pa* for short) on the values of AC, PR, RE, and RI of the proposed CCS-K-Prototypes algorithm for clustering this dataset, respectively. From these figures, we can see that AC achieves its highest value when the number of nests N equals to 20, and the abandon probability *pa* equals to 0.3; PR obtains its highest value when the number of nests N equals to 20, and the abandon probability *pa* equals to 0.3; RE achieves its highest value when the number of nests N equals to 20, and the abandon probability *pa* equals to 0.3; RI obtains its highest value when the number of nests N equals to 20, and the abandon probability *pa* equals to 0.3. The results in Figs. 3a-3d show that the performance of the proposed CCS-K-Prototypes algorithm on heart disease dataset (second case) is affected by the number of nests N and the abandon probability *pa* not obviously except for *pa* equalling to 0.25, and the CCS-K-Prototypes algorithm can achieve reasonably good results within the given range of N and *pa*. In Tables 5a-5d, we summarize the clustering results of the CCS-K-Prototypes, the K-Prototypes, the SBAC, the KL-FCM-GM, the EKP, and the ABC-K-Prototypes algorithms on the heart diseases dataset (second case) according to AC, PR, RE, and RI, respectively. The results in Tables 5a-5d indicate that our proposed CCS-K-Prototypes approach achieves the highest value on all the four measures AC, PR, RE, and RI.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

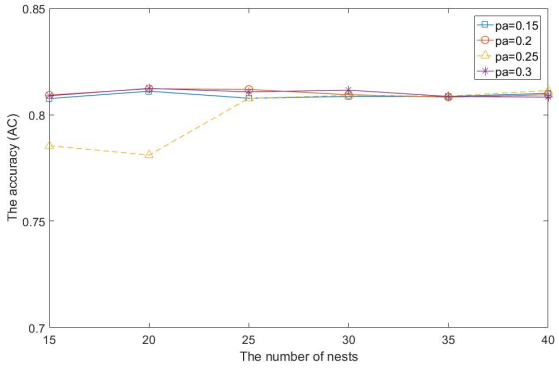

FIGURE 3a. The accuracy (AC) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the heart disease dataset (2 classes and 13 attributes)

TABLE 5a. The accuracy (AC) of the six algorithms on the heart disease dataset (2 classes and 13 attributes)

Algorithms	AC
K-Prototypes	0.577
SBAC	0.545
KL-FCM-GM	0.762 (α =1.7)
EKP	0.545
ABC-K-Prototypes	0.809
CCS-K-Prototypes	**0.812** (*N*=20, *pa*=0.3)

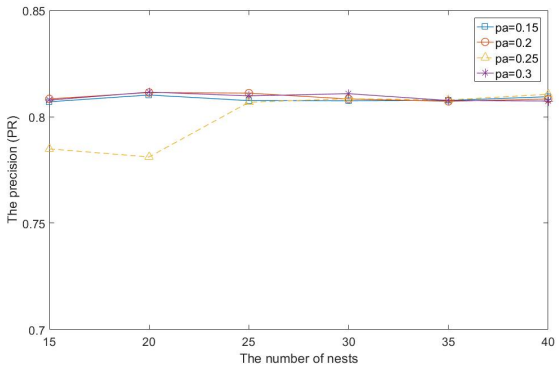

FIGURE 3b. The precision (PR) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the heart disease dataset (2 classes and 13 attributes)

TABLE 5b. The precision (PR) of the six algorithms on the heart disease dataset (2 classes and 13 attributes)

Algorithms	PR
K-Prototypes	0.570
SBAC	0.567
KL-FCM-GM	0.783 (α = 2.6)
EKP	0.272
ABC-K-Prototypes	0.808
CCS-K-Prototypes	**0.812** (*N*=20, *pa*=0.3)

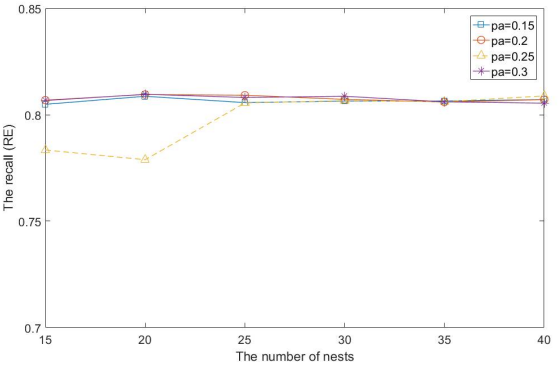

FIGURE 3c. The recall (RE) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the heart disease dataset (2 classes and 13 attributes)

TABLE 5c. The recall (RE) of the six algorithms on the heart disease dataset (2 classes and 13 attributes)

Algorithms	RE
K-Prototypes	0.566
SBAC	0.5
KL-FCM-GM	0.768 (α =1.7)
EKP	0.5
ABC-K-Prototypes	0.806
CCS-K-Prototypes	**0.809** (*N*=20, *pa*=0.3)

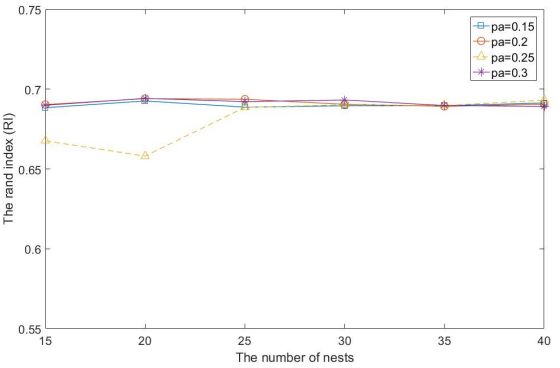

FIGURE 3d. The rand index (RI) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the heart disease dataset (2 classes and 13 attributes)

TABLE 5d. The rand index (RI) of the six algorithms on the heart disease dataset (2 classes and 13 attributes)

Algorithms	RI
K-Prototypes	0.510
SBAC	0.499
KL-FCM-GM	0.641 (α =1.7)
EKP	0.502
ABC-K-Prototypes	0.689
CCS-K-Prototypes	**0.694** (*N*=20, *pa*=0.3)

The credit approval dataset contains 690 data objects from credit card organizations, where each data object has ten categorical attributes and six numeric attributes. The last attribute is the class attribute, and has two values (negative

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

and positive). Figs. 4a-4d illustrate the effect of the number of nests N and the abandon probability pro_a (*pa* for short) on the values of AC, PR, RE, and RI of the proposed CCS-K-Prototypes algorithm for clustering this dataset, respectively. From these figures, we can see that AC achieves its highest value when the number of nests N equals to 30, and the abandon probability *pa* equals to 0.2; PR obtains its highest value when the number of nests N equals to 30, and the abandon probability *pa* equals to 0.2; RE achieves its highest value when the number of nests N equals to 30, and the abandon probability *pa* equals to 0.2; RI obtains its highest value when the number of nests N equals to 30, and the abandon probability *pa* equals to 0.2. The results in Figs. 4a-4d show that the performance of the proposed CCS-K-Prototypes algorithm on the credit approval dataset is affected by the number of nests N and the abandon probability *pa* not obviously, and the CCS-K-Prototypes algorithm can achieve reasonably good results within the given range of N and *pa*. In Tables 6a-6d, we summarize the clustering results of the CCS-K-Prototypes, the K-Prototypes, the SBAC, the KL-FCM-GM, the EKP, and the ABC-K-Prototypes algorithms on this dataset according to AC, PR, RE, and RI, respectively. In Tables 6a-6b, we also supply the clustering results of the algorithm ACC-FSFDP, which are taken from [21]. The results in Tables 6a-6d indicate that our proposed CCS-K-Prototypes approach achieves the highest values or near highest values on all the four measures AC, PR, RE, and RI.

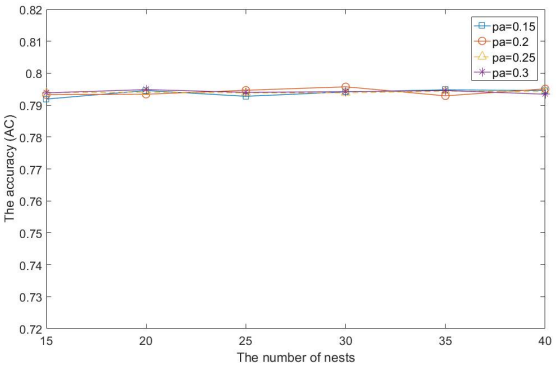

FIGURE 4a. The accuracy (AC) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the credit approval dataset

TABLE 6a. The accuracy (AC) of the seven algorithms on the credit approval dataset

Algorithms	AC
K-Prototypes	0.562
SBAC	0.555
KL-FCM-GM	0.578 ($\alpha = 2.4$)
EKP	0.686
ABC-K-Prototypes	0.794
ACC-FSFDP	0.784
CCS-K-Prototypes	**0.796** (N=30, pa=0.2)

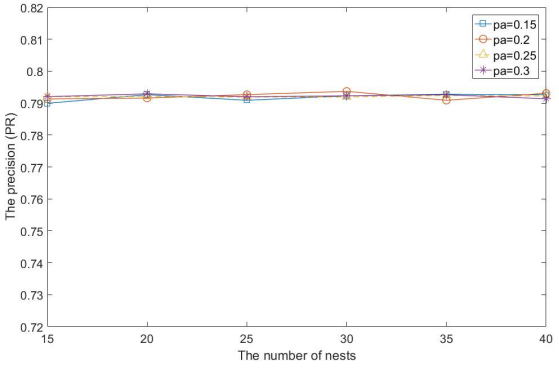

FIGURE 4b. The precision (PR) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the credit approval dataset

TABLE 6b. The precision (PR) of the seven algorithms on the credit approval dataset

Algorithms	PR
K-Prototypes	0.780
SBAC	0.558
KL-FCM-GM	0.642 ($\alpha = 2.4$)
EKP	0.724
ABC-K-Prototypes	0.792
ACC-FSFDP	**0.814**
CCS-K-Prototypes	0.794 (N=30, pa=0.2)

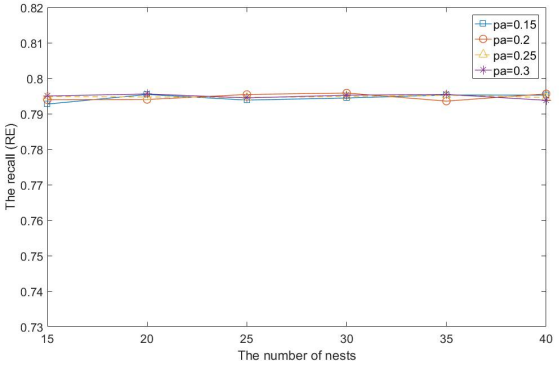

FIGURE 4c. The recall (RE) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the credit approval dataset

TABLE 6c. The recall (RE) of the six algorithms on the credit approval dataset

Algorithms	RE
K-Prototypes	0.508
SBAC	0.5
KL-FCM-GM	0.549 ($\alpha = 2.4$)
EKP	0.657
ABC-K-Prototypes	0.795
CCS-K-Prototypes	**0.796** (N=30, pa=0.2)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

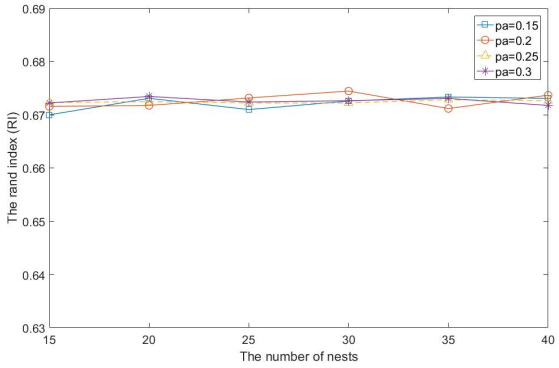

FIGURE 4d. The rand index (RI) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the credit approval dataset

TABLE 6d. The rand index (RI) of the six algorithms on the credit approval dataset

Algorithms	RI
K-Prototypes	0.507
SBAC	0.499
KL-FCM-GM	0.513 ($\alpha = 2.4$)
EKP	0.568
ABC-K-Prototypes	0.673
CCS-K-Prototypes	**0.674** (*N*=30, *pa*=0.2)

The soybean dataset consists of 47 data objects, where each data object is described by 36 categorical attributes. The last attribute is the class attribute with four values: diaporthe-stem-canker, charcoal-rot, rhizoctonia-root-rot, and phytophthora-rot. Figs. 5a-5d illustrate the effect of the number of nests N and the abandon probability pro_a (*pa* for short) on the values of AC, PR, RE, and RI of the proposed CCS-K-Prototypes algorithm for clustering this dataset, respectively. From these figures, we can see that AC, PR, RE, and RI achieve their highest values when the number of nests N equals to 20, and the probability *pa* equals to 0.3. The results in Figs. 5a-5d show that the performance of the proposed CCS-K-Prototypes algorithm on the soybean dataset is not significantly affected by the number of nests N and the abandon probability *pa*, and the CCS-K-Prototypes algorithm can achieve reasonably good results within the given range of N and *pa*. In Tables 7a-7d, we list the clustering results of the CCS-K-Prototypes, K-Prototypes, SBAC, KL-FCM-GM, EKP, and ABC-K-Prototypes algorithms on this dataset in terms of AC, PR, RE, and RI, respectively. In Tables 7a-7b, we also supply the clustering results of the algorithm ACC-FSFDP, which are taken from [21]. The results in Tables 7a-7d indicate that performance of our proposed CCS-K-Prototypes approach outperforms most other algorithms according to the measures of AC, PR, RE, and RI.

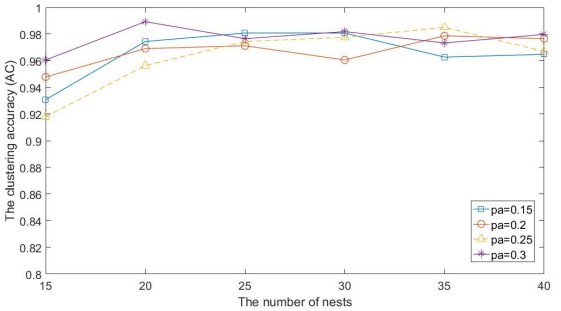

FIGURE 5a. The accuracy (AC) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the soybean dataset

TABLE 7a. The accuracy (AC) of the seven algorithms on the soybean dataset

Algorithms	AC
K-Prototypes	0.855
SBAC	0.362
KL-FCM-GM	0. 969 ($\alpha = 2.8$)
EKP	**0.993**
ABC-K-Prototypes	0.982
ACC-FSFDP	0.957
CCS-K-Prototypes	0.989 (*N*=20, *pa*=0.3)

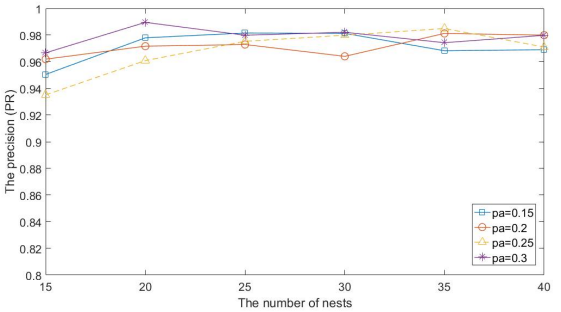

FIGURE 5b. The precision (PR) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the soybean dataset

TABLE 7b. The precision (PR) of the seven algorithms on the soybean dataset

Algorithms	PR
K-Prototypes	0.903
SBAC	0.829
KL-FCM-GM	0.983 ($\alpha = 2.8$)
EKP	**0.993**
ABC-K-Prototypes	0.983
ACC-FSFDP	0.985
CCS-K-Prototypes	0.990 (*N*=20, *pa*=0.3)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

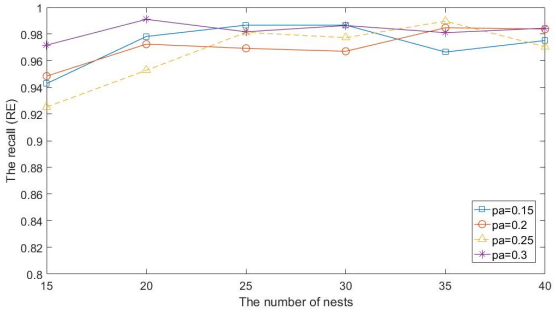

FIGURE 5c. The recall (RE) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the soybean dataset

TABLE 7c. The recall (RE) of the six algorithms on the soybean dataset

Algorithms	RE
K-Prototypes	0.849
SBAC	0.250
KL-FCM-GM	0.968 ($\alpha = 2.8$)
EKP	**0.994**
ABC-K-Prototypes	0.988
CCS-K-Prototypes	0.991 (*N*=20, *pa*=0.3)

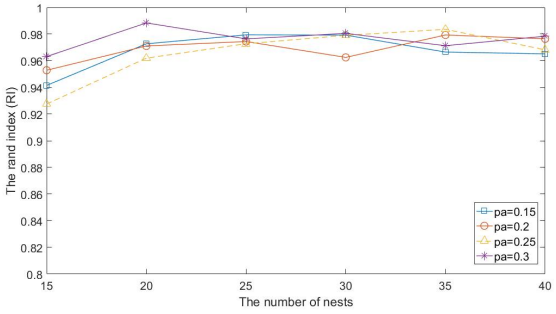

FIGURE 5d. The rand index (RI) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the soybean dataset

TABLE 7d. The rand index (RI) of the six algorithms on the soybean dataset

Algorithms	RI
K-Prototypes	0.884
SBAC	0.292
KL-FCM-GM	0.973 ($\alpha = 2.8$)
EKP	**0.993**
ABC-K-Prototypes	0.981
CCS -K-Prototypes	0.988 (*N*=20, *pa*=0.3)

The breast cancer dataset has 699 data objects, where each data object is described by eleven attributes. The first attribute is the sample code number, which is not used in clustering analysis; while the last attribute is the class attribute, and it has two values: benign and malignant. Figs. 6a-6d illustrate the effect of the number of nests *N* and the abandon probability pro_a (*pa* for short) on the AC, PR, RE, and RI of the proposed CCS-K-Prototypes algorithm for clustering this dataset, respectively. From these figures, we can see that AC, PR, RE, and RI achieve their highest values when the number of nests *N* equals to 40, and the abandon probability *pa* equals to 0.3. The results in Figs. 6a-6d show that the performance of the proposed CCS-K-Prototypes algorithm on breast cancer dataset is not significantly affected by the number of nests *N* and the abandon probability *pa*, and the CCS-K-Prototypes algorithm can achieve reasonably good results within the given range of *N* and *pa*. In Tables 8a-8d, we list the clustering results of the CCS-K-Prototypes, the K-Prototypes, the SBAC, the KL-FCM-GM, the EKP, and the ABC-K-Prototypes algorithms on this dataset according to AC, PR, RE, and RI, respectively. In Tables 8a-8b, we also supply the clustering results of the algorithm ACC-FSFDP, which are taken from [21]. The results in Tables 8a-8d show that performance of our proposed CCS-K-Prototypes approach outperforms most other algorithms according to the measures AC, PR, RE, and RI.

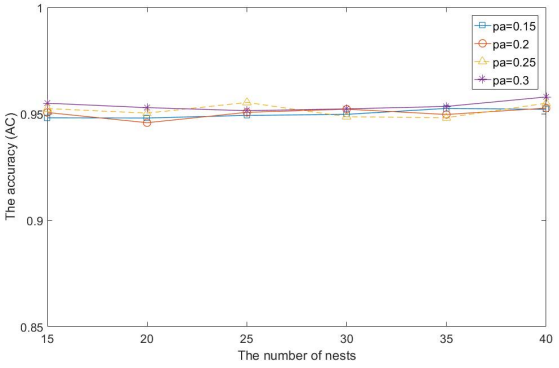

FIGURE 6a. The accuracy (AC) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the breast cancer dataset

TABLE 8a. The accuracy (AC) of the seven algorithms on the breast cancer dataset

Algorithms	AC
K-Prototypes	**0.961**
SBAC	0.655
KL-FCM-GM	0.804 ($\alpha = 1.1$)
EKP	0.701
ABC-K-Prototypes	0.959
ACC-FSFDP	0.938
CCS-K-Prototypes	0.958 (*N*=40, *pa*=0.3)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2973216, IEEE Access

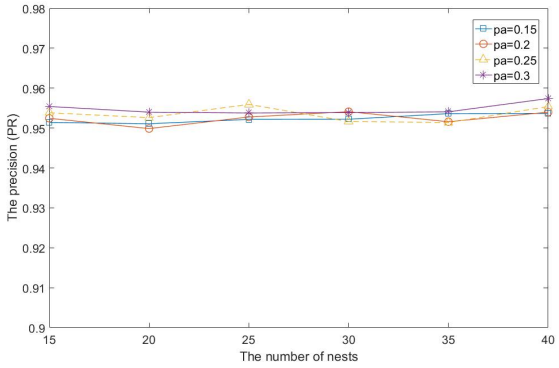

FIGURE 6b. The precision (PR) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the breast cancer dataset

TABLE 8b. The precision (PR) of the seven algorithms on the breast cancer dataset

Algorithms	PR
K-Prototypes	**0.959**
SBAC	0.650
KL-FCM-GM	0.813 ($\alpha = 1.1$)
EKP	0.767
ABC-K-Prototypes	0.958
ACC-FSFDP	0.947
CCS-K-Prototypes	0.957 (*N*=40, *pa*=0.3)

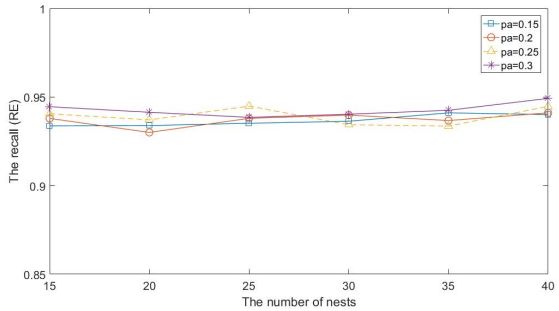

FIGURE 6c. The recall (RE) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the breast cancer dataset

TABLE 8c. The recall (RE) of the six algorithms on the breast cancer dataset

Algorithms	RE
K-Prototypes	**0.954**
SBAC	0.500
KL-FCM-GM	0.753 ($\alpha = 1.1$)
EKP	0.771
ABC-K-Prototypes	0.952
CCS-K-Prototypes	0.949 (*N*=40, *pa*=0.3)

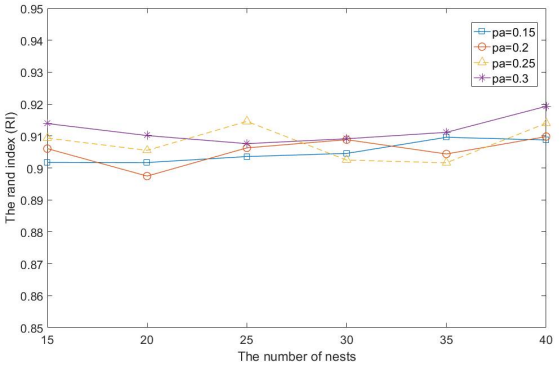

FIGURE 6d. The rand index (RI) of the CCS-K-Prototypes algorithm with varying values of *N* and *pa* on the breast cancer dataset

TABLE 8d. The rand index (RI) of the six algorithms on the breast cancer dataset

Algorithms	RI
K-Prototypes	**0.925**
SBAC	0.511
KL-FCM-GM	0.686 ($\alpha = 1.1$)
EKP	0.580
ABC-K-Prototypes	0.922
CCS-K-Prototypes	0.919 (*N*=40, *pa*=0.3)

The results in Figs. 1a-6d show that the number of nests *N* and the abandon probability pro_a (*pa* for short) have a slight impact on the performance of the proposed CCS-K-Prototypes approach on all datasets according to the measures of AC, PR, RE, and RI. Moreover, the CCS-K-Prototypes approach achieves high and stable values of all these measures in most cases. Therefore, we can say that the proposed CCS-K-Prototypes algorithm can achieve a reasonably good performance within the given range of *N* and *pa*. The experimental results in Tables 3a-8d show that our proposed CCS-K-Prototypes algorithm achieves the highest or near highest values of AC, PR, RE, and RI in most cases. Therefore, the proposed CCS-K-Prototypes algorithm outperforms the other six popular algorithms according to these measures.

We believe that the reason for the success of the CCS-K-Prototypes approach in the above experiments is as follows: this approach has the ability of efficiently performing global search and local search by introducing the CS search framework. The global search is used to search for candidate solutions in the entire attribute space, and the local search is used to search for candidate solutions around existing solutions. Therefore, the proposed CCS-K-Prototypes algorithm can obtain optimal or near optimal results.

V. CONCLUSIONS AND FUTURE WORK

Data objects with both numeric and categorical attributes are ubiquitous in many real-world applications. In this paper, we have proposed a novel clustering algorithm CCS-K-Prototypes (Clustering based on Cuckoo Search and K-Prototypes) for mixed numeric and categorical data. In our

algorithm, we converted the clustering task into the problem of searching for the cluster centres. To deal with different types of attributes, we propose a representation method for candidate solutions, and develop approaches of local search around an existing solution and global search in the entire attribute space. These are the major innovations in this research. We then analysed the time and space complexity of the CCS-K-Prototypes algorithm, and tested this algorithm on five datasets according to the clustering accuracy (AC), precision (PR), recall (RE), and rand index (RI). We also illustrate the effect of the number of nests N and the abandon probability pro_a on the performance of the CCS-K-Prototypes algorithm. The results in Figs. 1a-6d show that the number of nests N and the abandon probability pro_a (pa for short) have a slight impact on the performance of the CCS-K-Prototypes algorithm. In addition, the CCS-K-Prototypes algorithm can achieve promising results on all datasets within the given range of the number of nests N and the abandon probability pro_a. In comparison with other six popular algorithms, we found that our proposed CCS-K-Prototypes algorithm outperformed other algorithms in most cases.

Our future work will consider sparse representation and multi-objective optimisation approaches to clustering mixed data. This is based on the following considerations: the sparse representation models can be used to obtain the sparse representation for a data object. Sparse representation has been applied in numerous computer vision tasks such as image classification and clustering, and it achieves promising performance [40], [41]. Additionally, multi-objective optimization via meta-heuristic has demonstrated promising results in many applications including clustering analysis [27], [42]. Therefore, in our future work, we would like to explore the potential of these two approaches to clustering mixed data.

Another line of future research will involve multi-view clustering and deep clustering for mixed data. Multi-view clustering was introduced by Bickel and Scheffer [43], and has attracted considerable attention in recently years [44]. With the success of deep learning, deep clustering is emerging as an interesting research direction. In [44], MvSCN (multi-view spectral clustering network) was proposed as a deep version of multi-view spectral clustering [44]. In addition, MMFA (multiple marginal Fisher analysis) can estimate the feature dimension automatically [45]. These approaches have demonstrated promising results on image and text data. Therefore, in our future work, we would like to explore the task of clustering the mixed data by integration these two approaches with CS strategy.

REFERENCES

[1] M. Celebi, H. Kingravi, and P. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Systems with Applications,* vol. 40, no. 1, pp. 200-210, Jan, 2013.

[2] A. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters,* vol. 31, no. 8, pp. 651-666, Jun, 2010.

[3] C. Luo, W. Pang, and Z. Wang, "Semi-supervised clustering on heterogeneous information networks," in *Proc. of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'14)*, Taiwan, 2014, pp. 548-559.

[4] G. Bordogna, and G. Pasi, "A quality driven hierarchical data divisive soft clustering for information retrieval," *Knowledge-Based Systems,* vol. 26, pp. 9-19, Feb, 2012.

[5] F. Naouar, L. Hlaoua, and M. Omri, "Collaborative information retrieval model based on fuzzy clustering," in *Proc. of International Conference on High Performance Computing & Simulation (HPCS)*, Genoa, Italy, 2016, pp. 495-502.

[6] C. Bogner, B. Widemann, and H. Lange, "Characterising flow patterns in soils by feature extraction and multiple consensus clustering," *Ecological Informatics,* vol. 15, pp. 44-52, May, 2013.

[7] Y. Xin, Z. Xie, and J. Yang, "The privacy preserving method for dynamic trajectory releasing based on adaptive clustering," *Information Sciences,* vol. 378, pp. 131-143, Feb, 2017.

[8] K. Bharti, and P. Singh, "Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering," *Applied Soft Computing,* vol. 43, pp. 20-34, Jun, 2016.

[9] F. Saeed, N. Salim, and A. Abdo, "Information theory and voting based consensus clustering for combining multiple clusterings of chemical structures," *Molecular Informatics,* vol. 32, no. 7, pp. 591-598, May, 2013.

[10] P. Blomstedt, R. Dutta, S. Seth, A. Brazma, and S. Kaski, "Modelling-based experiment retrieval: a case study with gene expression clustering," *Bioinformatics,* vol. 32, no. 9, pp. 1388-1394, May, 2016.

[11] J. Ji, W. Pang, C. Zhou, X. Han, and Z. Wang, "A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data," *Knowledge-Based Systems,* vol. 30, pp. 129-135, Jun, 2012.

[12] J. Han, M. Kamber, and J. Pei, *Data mining concepts and techniques*, 3rd ed., SF, USA: Morgan Kaufmann, 2012, pp. 443-490.

[13] A. Jain, and R. Dubes, "Algorithms for clustering data," *Technometrics*, vol. 32, no. 2, pp. 227-229, 1988.

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

[15] Z. Huang, "Clustering large data sets with mixed numeric and categorical values." in *Proc. of the first Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1997, pp. 21-34.

[16] J. Bezdek, J. Keller, R. Krisnapuram, and N. Pal, *Fuzzy models and algorithms for pattern recognition and image processing*, Boston, USA: Springer, 1999, pp. 11-182.

[17] A. Ahmad, and L. Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data & Knowledge Engineering,* vol. 63, no. 2, pp. 503-527, Nov, 2007.

[18] J. Ji, T. Bai, C. Zhou, C. Ma, and Z. Wang, "An improved k-prototypes clustering algorithm for mixed numeric and categorical data," *Neurocomputing,* vol. 120, pp. 590-596, Nov, 2013.

[19] S. Chatzis, "A fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional," *Expert Systems with Applications,* vol. 38, no. 7, pp. 8684-8689, Jul, 2011.

[20] D. Lam, M. Wei, and D. Wunsch, "Clustering data of mixed categorical and numerical type with unsupervised feature learning," *IEEE Access,* vol. 3, pp. 1605-1613, Sep, 2015.

[21] J. Chen, and H. He, "A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data," *Information Sciences,* vol. 345, pp. 271-293, Jun, 2016.

[22] X. Yang, *Nature-Inspired Metaheuristic Algorithms*, Frome, UK: Luniver Press, 2010, pp. 1-108.

[23] D. Teodorović, "Bee colony optimization (BCO)," in *Innovations in Swarm Intelligence*, vol. 248, C. Lim, L. Jain and S. Dehuri, ed., Heidelberg, Germany: Springer, 2009, pp. 39-60.

[24] X. Yang, and S. Deb, "Cuckoo Search via Lévy flights." in *Proc. of World Congress on Nature & Biologically Inspired Computing*, 2009, pp. 210-214.

[25] X. Yang, and S. Deb, "Engineering Optimisation by Cuckoo Search," *International Journal of Mathematical Modelling & Numerical Optimisation,* vol. 1, no. 4, pp. 330-343, 2010.

[26] X. Yang, and S. Deb, "Cuckoo search: recent advances and applications," *Neural Computing & Applications,* vol. 24, no. 1, pp. 169-174, 2014.
[27] X. Yang, and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers & Operations Research,* vol. 40, no. 6, pp. 1616-1624, 2013.
[28] M. Marichelvam, T. Prabaharan, and X. Yang, "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan," *Applied Soft Computing,* vol. 19, no. 1, pp. 93-101, 2014.
[29] M. Adnan, M. Razzaque, M. Abedin, S. Reza, and M. Hussein, "A novel cuckoo search based clustering algorithm for wireless sensor networks." in *Proc. of ICOCOE*, Cham, Switzerland, 2016, pp. 621-634.
[30] S. Goel, A. Sharma, and P. Bedi, "Cuckoo search clustering algorithm: a novel strategy of biomimicry." in *Proc. of in 2011 World Congress Information & Communication Technologies*, Mumbai, India, 2011, pp. 916-921.
[31] J. Senthilnath, V. Das, S. Omkar, and V. Mani, "Clustering using levy flight cuckoo search," in *Proc. of 7th International Conference on Bio-Inspired Computing: Theories and Applications*, Gwalior, India, 2012, pp. 65-75.
[32] K. Lakshmi, N. Visalakshi, S. Shanthi, and S. Parvathavarthini, "Clustering categorical data using k-modes based on cuckoo search optimization algorithm " *ICTACT JOURNAL ON SOFT COMPUTING,* vol. 08, no. 01, pp. 1561-1566, Oct, 2017.
[33] K. Lakshmi, N. Visalakshi, and S. Shanthi, "Cuckoo search based k-prototype clustering algorithm," *Asian Journal of Research in Social Sciences and Humanities,* vol. 7, no. 2, pp. 300-309, Feb, 2017.
[34] Y. Yang, "An evaluation of statistical approaches to text categorization," *Journal of Information Retrieval,* vol. 1, pp. 67-88, 1999.
[35] W. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846-850, 1971.
[36] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery,* vol. 2, no. 3, pp. 283-304, 1998.
[37] C. Li, and G. Biswas, "Unsupervised learning with mixed numeric and nominal data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 14, no. 4, pp. 673-690, 2002.
[38] Z. Zheng, M. Gong, J. Ma, L. Jiao, and Q. Wu, "Unsupervised evolutionary clustering algorithm for mixed type data," in *Proc. of IEEE Congress on Evolutionary Computation*, Barcelona, Spain, 2010, pp. 1-8.
[39] J. Ji, Y. Chen, G. Feng, X. Zhao, F. He, "Clustering mixed numeric and categorical data with artificial bee colony strategy," *Journal of Intelligent & Fuzzy Systems,* vol. 36, pp. 1521-1530, Jan, 2019.
[40] X. Lan, M. Ye, R. Shao, B Zhong, P. Yuen, and H. Zhou, "Learning modality-consistency feature templates: a robust RGB-infrared tracking system," *IEEE Transactions on Industrial Electronics,* vol. 66, no. 12, pp. 9887-9897, 2019.
[41] E. Ehsan, and V. René, "Sparse subspace clustering: algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* vol. 35, no. 11, pp. 2765-2781, 2013.
[42] W. Rui, S. Lai, G. Wu, L. Xing, W. Ling, and H. Ishibuchi, "Multi-clustering via evolutionary multi-objective optimization," *Information Sciences,* vol. 450, pp. 128-140, Jun, 2018.
[43] S. Bickel, and T. Scheffer, "Multi-view clustering." in *Proc. of the 4th IEEE International Conference on Data Mining (ICDM'04)*, Brighton, UK, 2004, pp. 19-26.
[44] Z. Huang, J. Zhou, X. Peng, C. Zhang, H. Zhu, and J. Lv, "Multi-view spectral clustering network," in *Proc. of the 28th International Joint Conference on Artificial Intelligence*, Macao, China, 2019, pp. 2563-2569.
[45] Z. Huang, H. Zhu, J. Zhou, and X. Peng, "Multiple marginal fisher analysis," *IEEE Transactions on Industrial Electronics,* vol. 66, no. 12, pp. 9798-9807, 2019.