A hybrid kernel-based possibilistic fuzzy c-means clustering and cuckoo search algorithm

Viet Duc Do
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
e-mail: ducdoviet@gmail.com

Dinh Sinh Mai
Institute of Techniques for Special Engineering
Le Quy Don Technical University
Hanoi, Vietnam
e-mail: maidinhsinh@lqdtu.edu.vn

Long Thanh Ngo
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
e-mail: ngotlong@mta.edu.vn

***Abstract*—The possibilistic Fuzzy c-means (PFCM) algorithm is a robustness clustering algorithm which combines two algorithms, Fuzzy c-means (FCM) and Possibilistic c-means (PCM). It deals with the weaknesses of FCM in handling noise sensitivity and the weaknesses of PCM in the case of coincidence clusters. However, PFCM still has a common weakness of clustering algorithms. It can not separate nonlinearly separate clusters in input space, and their boundaries between two clusters are linear. To solving the nonlinear separable problem, kernel methods have been introduced into possibilistic fuzzy c-means clustering (KPFCM). KPFCM can deal with noises or outliers better than PFCM. But KPFCM suffers from a common weakness of clustering algorithms that may be trapped in a local minimum, leading to no good results. Recently, Cuckoo search (CS) based clustering has proved to achieve exciting results. It can achieve the best global solution compared to most other metaheuristics. This paper proposes a hybrid method encompassing KPFCM and Cuckoo search algorithm to form the proposed KPFCM-CSA. The experimental results show that the proposed algorithm achieved better clustering quality than some recent well-known clustering algorithms.**

Keywords*—*Possibilistic fuzzy c-means, Kernel method, Cuckoo Search, Fuzzy clustering

I. Introduction

Clustering is an unsupervised classification technique of data mining [1, 2]. It divides a set of data into groups or clusters based on the similarity between the data objects, such that similar objects fall in the same cluster and different objects in different clusters. Clustering has been used for a variety of applications such as statistics, machine learning, data mining, pattern recognition, bioinformatics, image analysis [3, 4, 28].

There are two commonly used clustering methods: hard clustering and soft (fuzzy) clustering. K-means [5] is a typical hard clustering algorithm. That is, each data point belongs only to a single cluster. This method makes it difficult to handle data where the patterns can simultaneously belong to many clusters. While Fuzzy c-means (FCM) [6] is an algorithm that represents fuzzy clustering, the membership value indicates the possibility that the data sample will belong to a particular cluster. For each data sample, the sum of the membership degree is equal to 1, and the large membership degree represents the data sample closer to the cluster centroid. However, the FCM is shown to be sensitive to noise and outliers [6]. To overcome these disadvantages, Krishnapuram and Keller have presented the possibilistic c-means (PCM) algorithm [7] by abandoning the constraint of FCM and constructing a novel objective function. PCM can deal with noisy data better. But PCM is very sensitive to initializations and sometimes generates coincident clusters. PCM considers the possibility (typicality) but neglects the important membership.

After that, Pal et al. proposed the possibilistic fuzzy c-means (PFCM) [8] algorithm with the assumption that membership and typicality are both important for accurate clustering. It is a combination of two algorithms FCM and PCM. PFCM algorithm deals with the weaknesses of FCM in handling noise sensitivity and the weaknesses of PCM in the case of coincidence clusters [29, 30]. However, it is observed that PFCM tends to give not-so-good results for unequal-sized clusters. To improve this algorithm, Tushir et al. [9] propose a new Kernel-based hybrid c-means (KPFCM) clustering model, which adopts a Kernel induced metric in the data space to replace the original Euclidean norm metric. By replacing the inner product with an appropriate 'Kernel' function, one can implicitly perform a non-linear mapping to a high dimensional feature space in which the data is more clearly separable [26, 27]. The proposed method is characterized by higher clustering accuracy than the original PFCM.

Recently, nature-inspired approaches have received increased attention from researchers dealing with data clustering problems [10]. In order to improve the KPFCM algorithm, we propose in this paper to use a new metaheuristic approach. It is mainly based on the cuckoo search (CS) algorithm, which was proposed by Xin-She Yang and Suash Deb in 2009 [11, 12]. CS is a search method that imitates obligate brood parasitism of some female cuckoo species specializing in mimicking the color and pattern of few chosen host birds. The parasitic cuckoo often chooses a nest where the host has just laid its own eggs so that when the firstly cuckoo chick hatches, it evicts the host eggs out of the nest to increase its own food share. Specifically, from an optimization standpoint, CS (i) can achieve global convergence, (ii) has local and global search capabilities controlled via a switching parameter (pa), and (iii) uses Levy flights rather than standard random walks to scan the design space more efficiently than the simple Gaussian process [13, 14]. In addition, the CS algorithm has the advantages of simple structure, few input parameters, easy realization, random search path, and robust for many optimization problems [15, 16] and its superiority in benchmark comparisons [17, 18] against particle swarm optimization (PSO) and genetic algorithm (GA) makes it an intelligent choice. In this paper, a hybrid kernel-based possibilistic fuzzy c-means (KPFCM) clustering and Cuckoo search algorithm is proposed and compared. The efficiency of the proposed algorithm is tested on five different data sets issued from the UCI Machine Learning Repository, and the obtained results are compared with some recent well-known clustering algorithms.

The remainder of this paper is organized as follows. Section II briefly introduces some background about PFCM,

978-1-6654-0435-8/21/$31.00 ©2021 IEEE

KPFCM, and Cuckoo search algorithm (CSA). Section III proposes a hybrid algorithm of KPFCM and CSA. Section IV offers some experimental results, and Section V draws conclusions and suggests future research directions.

II. Background

A. Possibilistic fuzzy c-means clustering

Possibilistic Fuzzy c-means (PFCM) algorithm is a powerful clustering algorithm. PFCM overcomes the problem of noise of FCM and coincident cluster problem of PCM. It is a blended version of FCM clustering and PCM clustering. The PFCM algorithm has two types of memberships: a possibilistic (t_{ik}) membership that measures the absolute degree of typicality of a point in any particular cluster and a fuzzy membership (μ_{ik}) that measures the relative degree of sharing of a point among the clusters. Given a dataset $X = \{x_k\}_{k=1}^{n} \in R^M$, the PFCM finds the partition of X into $1 < c < n$ fuzzy subsets by minimizing the following objective function:

$$J_{m,\eta}(U,T,V) = \sum_{i=1}^{c}\sum_{k=1}^{n}(au_{ik}^{m} + bt_{ik}^{\eta})d_{ik}^{2} + \sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-t_{ik})^{\eta} \quad (1)$$

Where $U = \left[\mu_{ik}\right]_{c\times n}$ is a fuzzy partition matrix that contains the fuzzy membership degree; $T = \left[t_{ik}\right]_{c\times n}$ is a typicality partition matrix that contains the possibilistic membership degree; $V = (v_1, v_2, ..., v_c)$ is a vector of cluster centers, *m* is the weighting exponent for the fuzzy partition matrix, η is the weighting exponent for the typicality partition matrix, $\gamma_i > 0$ are constants given by the user and d_{ik}^{2} is the distance between the data points. The constants *a* and *b* define the relative importance of the membership and typicality values, respectively.

The PFCM model is subject to the following constraints:

$$\sum_{i=1}^{c}u_{ik} = 1; \sum_{k=1}^{n}t_{ik} = 1; 1 \le i \le c; 1 \le k \le n \quad (2)$$

$$a > 0, b > 0, m > 1, \eta > 1, 0 \le \mu_{ik}, t_{ik} \le 1 \quad (3)$$

The objective function reaches the smallest value with constraints (2) and (3) when it follows condition:

$$\mu_{ik} = 1 / \sum_{j=1}^{c}\left(d_{ik}^{2} / d_{jk}^{2}\right)^{1/(m-1)} \quad (4)$$

$$\gamma_i = K\sum_{k=1}^{n}u_{ik}^{m}d_{ik}^{2} / \sum_{k=1}^{n}u_{ik}^{m} \quad (5)$$

Typically, K is chosen as 1.

$$t_{ik} = 1 / \left(1 + \left(bd_{ik}^{2} / \gamma_i\right)^{1/(\eta-1)}\right) \quad (6)$$

$$v_i = \frac{\sum_{k=1}^{n}(au_{ik}^{m} + bt_{ik}^{\eta})x_k}{\sum_{k=1}^{n}(au_{ik}^{m} + bt_{ik}^{\eta})} \quad (7)$$

The PFCM algorithm can be summarized as follows.

Algorithm 1: Possibilistic Fuzzy C-means Algorithm

```
Input: Dataset $X = \{x_k\}_{k=1}^{n} \in R^M$, the number of clusters c ($1 < c < n$), fuzzifier parameters a, b, m, $\eta$, stop condition $T_{\max}$, $\varepsilon$ ; and $t = 0$.
Output: The membership matrix U, T and the centroid matrix V.
Step 1: Initialize the centroid matrix $V^{(0)}$ by choosing randomly from the input dataset X.
Step 2: Repeat
    2.1 $t = t + 1$
    2.2 Compute matrix $U^{(t)}$ by using Eq. (4)
    2.3 Compute typical $\gamma_i$ by using Eq. (5)
    2.4 Compute matrix $T^{(t)}$ by using Eq. (6)
    2.5 Update the centroid $V^{(t)}$ by using Eq. (7)
    2.6 Check if $\left\|V^{(t)} - V^{(t-1)}\right\| \le \varepsilon$ or $t > T_{\max}$ .If yes
then stop and go to Output, otherwise return Step 2.
```

B. Kernel-based possibilistic fuzzy c-means clustering

The possibilistic fuzzy c-means model uses the Euclidean distance to calculate the fuzzy memberships by Eq. (4). However, in the real world, the Euclidean distance is not complex enough to deal with a more complex problem. Here, we use kernel methods to calculate the distance. Through some nonlinear mapping, the input data are mapped implicitly into a high-dimensional feature space in which they are more clearly separable where a possibilistic fuzzy c-means algorithm is performed.

A KPFCM clustering was proposed in Tushir and Srivastava (2010) [9], which used Gaussian kernel in the induced distance metric. KPFCM algorithm basically adopts a Kernel-induced metric different from the Euclidean norm in the original PFCM. The KPFCM model minimizes the following objective function:

$$J_{KPFCM}(U,T,V) = \sum_{i=1}^{c}\sum_{k=1}^{n}(au_{ik}^{m} + bt_{ik}^{\eta}) \| \Phi(x_k) - \Phi(v_i) \|^2 + \sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-t_{ik})^{\eta} \quad (8)$$

Where $\| \Phi(x_k) - \Phi(v_i) \|^2$ is the square of distance between $\Phi(x_k)$ and $\Phi(v_i)$. The distance in the feature space is calculated through the Kernel in the input space as follows:

$$\| \Phi(x_k) - \Phi(v_i) \|^2 = K(x_k, x_k) - 2K(x_k, v_i) + K(v_i, v_i)$$

In this paper, we conducted the Gaussian kernel function which is used almost exclusively in the literature.

$$K(x_k, v_i) = \exp\left(-\frac{\| x_k - v_i \|^2}{2\sigma^2}\right); \sigma > 0$$

Then $K(x,x)=1$ and $\| \Phi(x_k)-\Phi(v_i)\|^2 = 2(1-K(x_k,v_i))$. Thus, the objective function (8) is transformed into:

$$J_{KPFCM}(U,T,V) = 2\sum_{i=1}^{c}\sum_{k=1}^{n}(au_{ik}^{m}+bt_{ik}^{\eta})(1-K(x_k,v_i)) + \sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-t_{ik})^{\eta} \tag{9}$$

subject to the constraints $\sum_{i=1}^{c} u_{ik} = 1 \forall k$, and $0 \le u_{ik}^{m}, t_{ik}^{\eta} \le 1$. Here $a,b>0; m,n>1; \gamma_i > 0$ .Given the constraints, objective function $J_{KPFCM}(U,V,T)$ can be solved by using the Lagrange multiplier method in order to determine U, V, T as follows:

$$\mu_{ik} = 1 / \sum_{j=1}^{c}\left(\frac{1-K(x_k,v_i)}{1-K(x_k,v_j)}\right)^{1/(m-1)} \tag{10}$$

$$\gamma_i = 2K\sum_{k=1}^{n} u_{ik}^{m}(1-K(x_k,v_i)) / \sum_{k=1}^{n} u_{ik}^{m} \tag{11}$$

K is a user-defined constant (usually selected by 1).

$$t_{ik} = 1/\left(1+\left(2b(1-K(x_k,v_i))/\gamma_i\right)^{1/(\eta-1)}\right) \tag{12}$$

$$v_i = \frac{\sum_{k=1}^{n}(au_{ik}^{m}+bt_{ik}^{\eta})K(x_k,v_i)x_k}{\sum_{k=1}^{n}(au_{ik}^{m}+bt_{ik}^{\eta})K(x_k,v_i)} \tag{13}$$

Algorithm 2: Kernel-based Possibilistic Fuzzy C-means Algorithm (KPFCM)

```
Input: Dataset $X = \{x_k\}_{k=1}^{n} \in R^M$, the number of clusters c ($1< c < n$), fuzzifier parameters a, b, m, $\eta$, stop condition $T_{\max}, \varepsilon$; the Kernel $\sigma$ and $t=0$.
Output: The membership matrix U, T and the centroid matrix V.
Step 1: Execute a FCM clustering algorithm to find initial $V^{(0)}$
Step 2: Repeat
    2.1 $t = t+1$
    2.2 Compute matrix $U^{(t)}$ by using Eq. (10)
    2.3 Compute typical $\gamma_i$ by using Eq. (11)
    2.4 Compute matrix $T^{(t)}$ by using Eq. (12)
    2.5 Update the centroid $V^{(t)}$ by using Eq. (13)
    2.6 Check if $\left\|V^{(t)} - V^{(t-1)}\right\| \le \varepsilon$ or $t \ge T_{\max}$ .If yes then stop and go to Output, otherwise return Step 2.
```

C. Cuckoo Search Algorithm

Cuckoo Search algorithm (CS) is a metaheuristic search algorithm that has been proposed recently by Yang and Deb [11, 12]. The algorithm is inspired by the reproduction strategy of cuckoos. The CS algorithm effectively solves the optimization problem by simulating the parasitic parenting and Levy flight of the cuckoo. Parasitization refers to the cuckoo does not nest during breeding but laid its own eggs in other nests, with other birds to reproduce. The cuckoo will find hatching and breeding birds which is similar to their own self [19], and quickly spawn eggs while the bird is out. Cuckoos egg usually hatches quicker than the other eggs. When this happens, the foreign cuckoo will remove the non-hatched eggs from the nest by pushing the eggs out of the nest. This behavior is aimed at reducing the probability of the legitimate eggs from hatching.

In order to simplify the process of cuckoo parasitism in nature, the CS algorithm is based on three idealized rules:

1. Each cuckoo only has one egg at a time and chooses a parasitic bird nest for hatching by a random walk.

2. In the selected parasitic bird nest, only the best nest can be retained to the next generation.

3. The number of nests is fixed, and there is a probability that a host can discover an alien egg. If this happens, the host can either discard the egg or the nest, and this results in building a new nest in a new location.

In the above three idealized rules, the search for a new bird's nest location path is as follows:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda); i = 1,2,...,n \tag{14}$$

In which $x_i^{(t)}$ stands for the ith bird's nest position in the t generation, $\alpha(\alpha > 0)$ is the step size control, usually $\alpha = 1$. $Levy(\lambda)$ is Levy random search path, its expression is as follows:

$$Levy(\lambda) = t^{-\lambda}; 1 < \lambda < 3 \tag{15}$$

Cuckoo search algorithm is very effective for global optimization problems since it maintains a balance between local random walk and the global random walk. The balance between local and global random walks is controlled by a switching parameter $p_a \in [0,1]$. After the new solution is generated, some solutions are discarded according to a probability p_a, and then the corresponding new solution is generated by the way of random walks, and iteration is completed. The CS algorithm flows as follows:

Algorithm 3: Cuckoo Search Algorithm (CSA)

```
Input: Objective function $f(x), X = (x_1, x_2, ...., x_d)^T$
Output: Postprocess results and visualization.
Step 1: Generate initial population of n host nests $x_i (i = 1,2,...,n)$
Step 2: While ($t \le T_{\max}$) or (stop criterion)
    2.1 $t = t+1$
    2.2 Get a cuckoo randomly by Levy flights
            evaluate its quality/fitness Fi
    2.3 Choose a nest among n (say, j) randomly
        If ($F_i \le F_j$) then replace j by the new solution
    2.4 A fraction ( $p_a$ ) of worse nests are
            abandoned and new ones are built;
    2.5 Keep the best solutions
            (or nests with quality solutions);
    2.6 Rank the solutions and find the current best.
```

III. Hybrid Kernel Possibilistic Fuzzy C-Means Clustering and Cuckoo Search Algorithm

In this study, we propose an algorithm called KPFCM-CSA, which is combined the Cuckoo search algorithm presented in this thesis with the kernel-based PFCM clustering algorithm. Similar to the KPFCM algorithm, it is necessary to define an objective function for the KPFCM-CSA algorithm. The hybrid algorithm between KPFCM and CSA is considered to be the following objective function:

$$F_{KPFCM-CSA}(U,T,V)=\frac{J_{KPFCM}(U,T,V)}{\min\limits_{i,j=1,...,c;i\#j}\|\Phi(v_i)-\Phi(v_j)\|} \qquad (16)$$

Where $v_i, v_j \in R^M$ for $1 \le i,j \le c; i \ne j$ is an estimated vector of cluster centers. We adopt the Gaussian function as a Kernel function, thus Eq. (16) can be written as:

$$F_{KPFCM-CSA}(U,T,V)=\frac{J_{KPFCM}(U,T,V)}{\min\limits_{i,j=1,...,c;i\#j} 2(1-K(v_i,v_j))} \qquad (17)$$

For solving the data clustering problem, the standard cuckoo search algorithm is adapted to reach the centroids of the clusters. For doing this, we suppose that we have *n* objects, and each object is defined by *m* attributes. In this work, the main goal of the CSA is to find *c* centroids of clusters which minimize the fitness function (17). In the CSA mechanism, the solutions are the nests and each nest is represented by a matrix (*c,m*) with *c* rows and *m* columns, where, the matrix rows are the centroids of clusters. After CSA was conducted, the best solution was the best centroids which the fitness function (17) reached the minimum value.

The steps to implement hybrid algorithm between KPFCM and CSA are as follows:

```
Algorithm 4: KPFCM-CSA Algorithm
Input: Dataset $X = \{x_k\}_{k=1}^{n} \in R^M$ , the number of
clusters c (1< c < n), fuzzifier parameters a, b, m, $\eta$ , stop
condition $T_{\max}$ ; the Kernel Gaussian $\sigma$ , number of
populations p, probability $p_a$ and t =0.
Output: $F_{Best}$ , $V_{Best}$ ,the membership matrix U, T.
Step 1: Initialization
    1.1 Initialize population of nests by using the FCM
        algorithm.
            $p_{nests} = \left[ V_j^{(0)} \right]; j = 1,..., p;$
            $V_j^{(0)} = \left[ v_i^{(0)} \right]; i = 1,..., c; V_j^{(0)} \in R^{CxM}$
    1.2 Calculate fitness of all nests by using Eqs. (10)
        - (12) and (17).
    1.3 Sort to find the best fitness $F_{Best}$ and it is also
        best centroids $V_{Best}$
Step 2: Hybrid algorithm of KPFCM and CSA
    2.1 t = t +1
    2.2 Generate new solution i by Eqs. (14) and (15).
    2.3 Calculate Fi by using Eqs. (10) - (12) and (17).
    2.4 Select random nest j (i#j).
            If (Fi < Fj) then Replace Fj by Fi
    2.5 Sort to keep the best fitness
    2.6 Generate a fraction $p_a$ of new solutions to
replace the worse nests by random. Calculate fitness of these
nests by Eqs. (10) - (12) and (17).
    2.7 Sort to find the best fitness $F_{Best}$ , $V_{Best}$
    2.8 Check If ( $t > T_{\max}$ ) then go to Step 3,
otherwise return Step 2.
Step 3: Compute matrix
    3.1 Compute matrix $U^{(t)}$ by using Eq. (10)
    3.2 Compute typical $\gamma_i$ by using Eq. (11)
    3.3 Compute matrix $T^{(t)}$ by using Eq. (12)
```

The KPFCM-CSA algorithm will perform iterations until the fitness function $F_{KPFCM-CSA}(U,T,V)$ reaches the minimum value, and the computational complexity of this algorithm with $T_{\max}$ is $O\left((p+6)T_{\max}Mnc\right)$.

IV. Experimental Results and Discussions

A. Dataset description

In this section, we perform several experiments to verify the performance of the proposed algorithms. The experiments were tested on the five datasets from the UCI Machine Learning Repository. All the five datasets from UCI that we employ in our experiments are famous databases that can easily take it is at https://archive.ics.uci.edu/ml/index.php. In Table 1, we describe the typical features of the datasets include iris, wine, seeds, breast cancer, and digits datasets.

TABLE I. THE CHARACTERISTICS OF THE TEST DATASETS

Dataset	Number of Instances	Number of Features	Number of Clusters
Iris	150	4	3
Wine	178	13	3
Seeds	210	7	3
Breast Cancer	569	32	2
Digits	5620	64	10

B. Parameter initialization and evaluation methods

In order to verify the feasibility of the proposed approach, experimental algorithms include FCM [6], PFCM [8], KPFCM [9] and KPFCM-CSA. The algorithms are executed for a maximum of 500 iterations and $\varepsilon = 10^{-6}$. For all algorithms, we first ran FCM algorithm with m =2 to determine the initial centroids. With the algorithms PFCM, KPFCM and KPFCM-CSA, K= 1 was selected to calculate the value γ_i by using Eqs. (5) and (11). The parameters of the PFCM algorithm were selected from [8]. The parameters of the KPFCM, KPFCM-CSA algorithm were selected as follows: $a = b = 1$, $m = n = 2$ and parameter σ has been chosen suitable for each dataset. In the KPFCM-CSA algorithm, the population size, step size, probability were selected from [11] specifically as follows: $p=15$, $\alpha = 0.01$, $p_a = 0.25$.

To assess the performance of algorithms, we use the following evaluation indicators as follows: Bezdek partition coefficient index (PC-I) [21], Dunn separation index (D-I), the classification entropy index (CE-I) [22], the Xie-Beni index (XB-I) [20], the mean squared error index (MSE) [23] and Davies Bouldin index [24]. Large values for indexes PC-I and D-I are good for clustering results, while small values for indexes CE-I, XB-I, DB-I and MSE are good for clustering results. Furthermore, the clustering results were measured using the accuracy measure *r* defined in [25]. The higher value of accuracy measure *r* proves superior clustering results with perfect clustering generating a value *r* = 1.

C. *Results and discussion*

We have implemented clustering on the different algorithms such as FCM, PFCM, KPFCM and KPFCM-CSA on five datasets. The experimental results are shown in some tables from II to VI. The clustering result obtained on the datasets Iris, Wine, Seeds, Breast Cancer, Digits is described in Table II, Table III, Table IV, Table V, Table VI, respectively.

TABLE II. INDEX EVALUATION OF ALGORITHMS FCM, PFCM, KPFCM AND KPFCM-CSA WITH IRIS DATASET (PARAMETER $\sigma = 0.175$)

Algorithm	D-I	PC-I	DB-I	MSE	CE-I	XB-I	Accuracy
FCM	0.0547	0.7425	0.7738	0.0475	1.9672	0.5762	0.8866
PFCM	0.0701	0.7639	0.7648	0.047	1.9145	0.5745	0.9133
KPFCM	0.0721	0.7749	0.7569	0.0464	1.7216	0.5662	0.9266
KPFCM-CSA	**0.0735**	**0.7761**	**0.7524**	**0.0461**	**1.7087**	**0.5515**	**0.9333**

TABLE III. INDEX EVALUATION OF ALGORITHMS FCM, PFCM, KPFCM AND KPFCM-CSA WITH WINE DATASET (PARAMETER $\sigma = 0.35$)

Algorithm	D-I	PC-I	DB-I	MSE	CE-I	XB-I	Accuracy
FCM	0.1413	0.7033	1.3181	0.2806	1.8546	0.4053	0.9454
PFCM	0.1423	0.7352	1.3181	0.2786	1.8258	0.3936	0.9494
KPFCM	0.1523	0.7694	1.3156	0.2785	1.7879	0.3892	0.9607
KPFCM-CSA	**0.1693**	**0.7699**	**1.3118**	**0.2698**	**1.7725**	**0.3878**	**0.9663**

TABLE IV. INDEX EVALUATION OF ALGORITHMS FCM, PFCM, KPFCM AND KPFCM-CSA WITH SEEDS DATASET (PARAMETER $\sigma = 0.75$)

Algorithm	D-I	PC-I	DB-I	MSE	CE-I	XB-I	Accuracy
FCM	0.0835	0.6915	0.8655	0.1067	1.3128	0.1985	0.8952
PFCM	0.0868	0.8196	0.8695	0.1066	1.2886	0.1977	0.8959
KPFCM	**0.0885**	0.8604	**0.8795**	0.1057	1.2296	0.1965	0.8995
KPFCM-CSA	**0.0885**	**0.8627**	**0.8795**	**0.1052**	**1.2266**	**0.1936**	**0.9095**

TABLE V. INDEX EVALUATION OF ALGORITHMS FCM, PFCM, KPFCM AND KPFCM-CSA WITH BREAST CANCER DATASET (PARAMETER $\sigma = 0.75$)

Algorithm	D-I	PC-I	DB-I	MSE	CE-I	XB-I	Accuracy
FCM	0.0838	0.6981	1.1486	0.3824	1.2107	0.3119	0.9232
PFCM	0.0838	0.7395	1.1466	0.3814	1.1974	0.2994	0.9279
KPFCM	**0.0861**	0.8571	1.1458	0.3787	1.1425	0.2508	0.9332
KPFCM-CSA	**0.0861**	**0.8599**	**1.1442**	**0.3773**	**1.1363**	**0.2394**	**0.9379**

TABLE VI. INDEX EVALUATION OF ALGORITHMS FCM, PFCM, KPFCM AND KPFCM-CSA WITH DIGITS DATASET (PARAMETER $\sigma = 0.75$)

Algorithm	D-I	PC-I	DB-I	MSE	CE-I	XB-I	Accuracy
FCM	0.1059	0.115	5.2998	4.7702	4.7026	1.5381	0.3632
PFCM	0.1186	0.1826	4.0674	4.7751	4.4825	1.5134	0.3754
KPFCM	0.1197	0.1857	3.7679	4.6885	4.4607	1.4185	0.3977
KPFCM-CSA	**0.1238**	**0.1887**	**3.6399**	**4.4513**	**4.4542**	**1.3715**	**0.4691**

From the clustering results of the five datasets which are shown in some tables form II to VI, according to the properties of datasets which are described in Table I and Fig. 1, some conclusions are revealed as follows:

- It is apparent that in terms of validity measures D-I, PC-I, DB-I, MSE, CE-I and XB-I, performance of the proposed KPFCM-CSA is better for most of the datasets.
- Performance of the proposed KPFCM-CSA algorithm is also measured by the clustering accuracy r. Again, the proposed algorithm obtained the highest clustering accuracy score for all datasets. The clustering accuracy obtained on the dataset Digits, Seeds, Iris, Breast Cancer, Wine are 46.91%, 90.95%, 93.33%, 93.79%, 96.63%, respectively.

- Fig. 1 describles the detailed clustering accuracy of all algorithms on five datasets. These results exhibit the KPFCM-CSA produces a better clustering solution than the other algorithms such as FCM, PFCM, and KPFCM.

From these, we can conclude that KPFCM-CSA can be the best clustering algorithm among the considered fuzzy clustering algorithms.

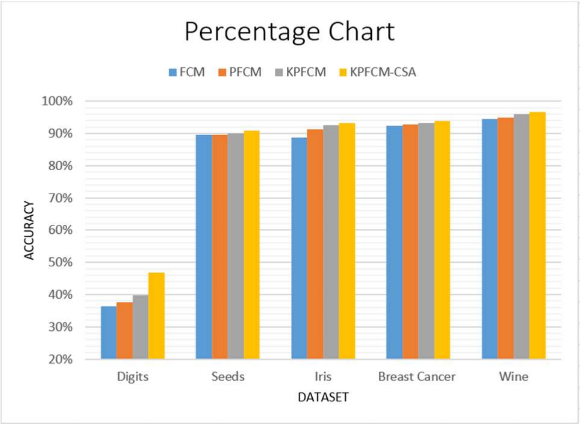

Fig. 1. The clustering accuracy of algorithms: FCM, PFCM, KPFCM and KPFCM-CSA.

V. CONCLUSION

The paper has proposed a hybrid algorithm between kernel-based PFCM and CSA. The experimental results show that the proposed method can achieve higher accuracy than some previous algorithms. According to the clustering results, when using some indicators to assess cluster quality, the KPFCM-CSA algorithm achieves the best results in most cases. Moreover, the kernel method used in the proposed algorithm can help to improve accuracy, improve stability while the CSA technique may avoid falling into local minima. In general, the KPFCM-CSA algorithm shows that it is a trustful, stable, accurate clustering algorithm and outperforms FCM, PFCM, and KPFCM.

In the future, we will develop a multiple kernel method based on PFCM to solve the complex problem of data and improve clustering accuracy. The development of optimization techniques to determine the suitable parameters for each dataset has also been a potential research direction.

REFERENCES

[1] Jain, K., Murthy, M.N., Flynn, P.J.(1999): Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323.

[2] Xu, R., Wunsch, D.C. (2009): Clustering, 2nd edn. IEEE Press, John Wiley and Sons, Inc, 1–13.

[3] I. H. Witten and E. Frank, Data Mining-Pratical Machine Learning Tools and Techniques, 3rd ed. Morgan Kaufmann Publishers, Inc., 2011.

[4] S. Mitra and T. Acharya (2003). Data Mining: Multimedia, Soft Computing, and Bioinformatics. Wiley.

[5] Anil K. Jain (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters; 31(8), 651-666.

[6] J. C. Bezdek (1981), Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.

[7] R. Krishnapuram and J. Keller (1993). A possibilistic approach to clustering. IEEE Trans. Fuzzy Systems, Vol. 1(2), 98–110.

[8] N. R. Pal, K. Pal, and J. C. Bezdek (2005). A possibilistic fuzzy c-means clustering algorithm," IEEE Trans. Fuzzy Systems, Vol. 13(4), 517–530.

[9] Tushir, M. and Srivastava, S. (2010). A new kernelized hybrid c-mean clustering with optimized parameters, Applied Soft Comp., Vol. 10, No. 3, pp.381–389, Elsevier.

[10] Colanzi, T.E., Assunção, W.K.K.G., Pozo, A.T.R., Vendramin, A.C.B.K., Pereira, D.A.B., Zorzo, C.A., de Paula Filho, P.L (2011). Application of Bio inspired Metaheuristics in the Data Clustering Problem. Clei Electronic Journal 14(3)..

[11] Yang, X.-S., Deb, S (2009). Cuckoo Search via Levy Flights. In: Proc. of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009). IEEE Publications, USA, 210–214.

[12] Yang, X.-S., Deb, S (2010). Engineering Optimisation by Cuckoo Search. International Journal of Mathematical Modelling and Numerical Optimisation 1(4-30), 330–343.

[13] M. Jamil, H.J. Zepernick, X.S. Yang (2013). Levy Flight Based Cuckoo Search Algorithm for Synthesizing Cross-Ambiguity Functions. IEEE Military Communications Conference (Milcom), San Diego, CA, 823–828.

[14] X.-S. Yang (2014), Nature-inspired Optimization Algorithm, first ed. Elsevier, MA, USA.

[15] Jothi, R., Vigneshwaran, A (2012). An Optimal Job Scheduling in Grid Using Cuckoo Algorithm. International Journal of Computer Science and Telecommunications 3(2), 65–69.

[16] Noghrehabadi, A., Ghalambaz, M., Ghalambaz, M., Vosough, A (2011). A hybrid Power Series –Cuckoo Search Optimization Algorithmto Electrostatic Deflection of Micro Fixed-fixed Actuators. International Journal of Multidisciplinary Sciences and Engineering, Vol. 2(4), 22–26.

[17] L.D. Coelho, C.E. Klein, S.L. Sabat, V.C. Mariani (2014). Optimal chiller loading for energy conservation using a new differential cuckoo search approach. Energy. Vol.75(1), 237–243.

[18] A. Natarajan, S. Subramanian, K. Premalatha (2012). A comparative study of cuckoo search and bat algorithm for Bloom filter optimisation in spam filtering. Int. J. Bio-Inspir. Comp. Vol.4(2), 89–99.

[19] Liyu, Mliang (20120. New Meta-heuristic Cuckoo Search Algorithm. Systems engineering. Vol.08, 64-69.

[20] U. Maulik, S. Bandyopadhyay (2002). Performance evaluation of some clustering algorithms and validity indices, IEEE Trans. Pattern Anal. Mach. Intell. Vol.24(12), 1650–1654.

[21] J.C. Bezdek, N. Pal (1998). Some new indexes of cluster validity, IEEE Trans. Syst. Man Cybern. Vol.28(3), 301–315.

[22] C.H. Chou, M.C. Su, E. Lai (2004). A new cluster validity measure and its application to image compression, Pattern Anal. Appl. Vol.7(2), 205–220.

[23] Z. Wang, A.C. Bovik (2009). Mean squared error: love it or leave it? A new look at signal fidelity measures, IEEE Signal Process. Mag. 98–117

[24] J. Cao, Z. Wu, J. Wu, and H. Xiong (2013). SAIL: Summationbased incremental learning for informationtheoretic text clustering, IEEE Transactions on Cybernetics, Vol. 43(2), 570–584.

[25] Z. Huang, M.K. Ng (1999). A fuzzy k-modes algorithm for clustering categorical data, IEEE Trans. Fuzzy Syst. 7(4), 446–452.

[26] D.S Mai, L.T Ngo (2018). Multiple kernel approach to semi-supervised fuzzy clustering algorithm for land-cover classification, Engineering Applications of Artificial Intelligence, Vol.68, 205-213.

[27] T.H Dang, D.S Mai, L.T Ngo (2019). Multiple kernel collaborative fuzzy clustering algorithm with weighted super-pixels for satellite image land-cover classification, Engineering Applications of Artificial Intelligence, Vol.85, 85-98.

[28] D.S Mai, L.T Ngo, T.L Hung (2018). Satellite Image Classification based Spatial-Spectral Fuzzy Clustering Algorithm, ACIIDS 2018, Vol.10752, 505–518.

[29] D.S Mai, L.T Ngo, T.L Hung (2018). Advanced Semi-supervised Possibilistic Fuzzy C-means Clustering using Spatial-Spectral distance for Land-cover Classification, SMC 2018, 4375-4380.

[30] D.S. Mai, L.T Ngo, T.L. Hung, H. Hagras (2021). A Hybrid Interval Type-2 Semi-supervised Possibilistic Fuzzy c-Means Clustering and Particle Swarm Optimization for Satellite Image Analysis, Information Sciences, Vol.548, 398-422.