Iranian Journal of Fuzzy Systems

Volume 20, Number 5, (2023), pp. 171-187

Original paper

University of Sistan and Baluchestan

Improving the genetic algorithm in fuzzy cluster analysis for numerical data and its applications

D. Pham Toan[1] and T. Vo Van[2]

[1] *Faculty of Mechanical - Electrical and Computer Engineering, School of Technology, Van Lang University, Ho Chi Minh City, Vietnam*

[2] *College of Natural Science, Can Tho University, Can Tho City, Vietnam*

dinh.pt@vlu.edu.vn, vvtai@ctu.edu.vn

Abstract

This study proposes an automatic genetic algorithm in fuzzy cluster analysis for numerical data. In this algorithm, a new measure called the FB index is used as the objective function of the genetic algorithm. In addition, the algorithm not only determines the appropriate number of groups but also improves the steps of traditional genetic algorithm as crossover, mutation and selection operators. The proposed algorithm is shown the step by step throughout the numerical example, and can perform fast by the established Matlab procedure. The result from experiments show the superiority of the proposed algorithm when it overcomes the existing algorithms. Moreover, it has been applied in recognizing the image data, and building the fuzzy time series model. These show the potential of this study for many real applications of the different fields.

Keywords: Fuzzy clustering, genetic algorithm, image recognition, time series.

1 Introduction

In the information century, we have to deal with a huge amount of data every day. In this problem, cluster analysis has basis role. Clustering is to divide a set of objects to become groups so that these elements have similarity according to a certain characteristic [4, 8, 9, 18, 20, 22]. Clustering aims to reduce the size of the data set by grouping similar data items together. The main objects of clustering can be the numerical data, probability density function data and interval data. Clustering for probability density function data (CPF) has considered by many authors such as [8, 20] and [22]. Clustering for interval data (CID) has also received with many interesting results. The important researches for CID were given by [11, 15, 16, 24] and [30].

Clustering for numerical data (CND) has studied in the first time with many announced results both theory and application [7, 10, 17, 32]. Comparing to CPF and CID, CND is more commonly used at present. In our opinion, there are some reasons for this. As known, databases in reality are often discrete. Therefore, to apply the CPF, we must estimate the probability density functions (PDF). Although there are many improvements in recent years, it is still a problem without the final solution [28]. Moreover, the measurements to separate clusters for the PDF are more complex than the numerical data. There are two main drawbacks of CPF in comparison with CND [28]. For CID, we also do not apply popularly as CND because interval data do not stored much in reality. Although applied in image recognition in recent researches, CID had the disadvantages about time cost and error in performing.

In cluster analysis, there are two main directions called the traditional and genetic algorithms. The traditional algorithm (TA) used a certain measure to evaluate the similarity of elements, and it is also the criterion to build clusters. In addition, these algorithms only focus on the rough clustering, for example, Nguyen et al. [21] introduced a new hybrid model developed based on Hierarchical K-means clustering and Cubist algorithm. Ma et al. [19] proposed

Corresponding Author: T. Vo Van
Received: February 2023; Revised: June 2023; Accepted: September 2023.
https://doi.org/10.22111/IJFS.2023.7834

a novel density-based radar scanning clustering algorithm. They used a fast mean-shift algorithm with adaptive radius and active subsets to effectively locate the centers, reducing the computational time significantly and then employed the shape of the PDF of the distribution of distances between a selected point and the other points in the data set. Yu et al. [33] proposed the two-way clustering method based on an improved DBSCAN algorithm. They build the clusters so that each cluster is described by a pair of nested sets called lower bound and upper bound respectively, instead of a single set to express a cluster in two-way clustering.

Some other authors used the soft clustering method. It has usually been implemented using the fuzzy c-means algorithm (FCM) that is based on the iterative optimization of an objective function to minimize the variation of objects within clusters. Notably this is a valuable characteristic of fuzzy c-means method as real data tends to be inherently noisy. The FCM algorithm is a typical unsupervised one, and is also an important branch of clustering methods. Dunn first proposed the FCM algorithm, and its objective function is then extended by [5]. Compared with the hard c-means algorithm, the concept of the membership degree in the FCM algorithm shows the degree of the sample belongs to the cluster [24]. Some typical studies as [34] proposed a stratified sampling based clustering algorithm for large-scale data is proposed in this paper. It has basic steps as follows: (1) obtaining a number of representative samples from different strata with a stratified sampling scheme, which are formed by locality sensitive hashing technique, (2) partitioning the chosen samples into different clusters using the fuzzy -means clustering algorithm, (3) assigning the out-of-sample objects into their closest clusters via data labeling technique. Wang et al. [31] applied the fuzzy c-mean to segment images. They are used fast bilateral filter to acquire local spatial and intensity information and then finds membership linking is achieved by summing all membership degrees calculated from previous iteration within every cluster in squared logarithmic form as the denominator of objective function. In recent year, some researchers used the genetic algorithm in the fuzzy clustering problem. This is interesting application interested in many statisticians.

The genetic algorithm (GA) uses the natural selection process such as crossover, mutation and selection to find the optimal solution in clustering. It has been particularly interested because of its remarkable advantages. There have been three important problems that have been researched and improved in cluster analysis: (i) determine the suitable number of clusters, (ii) build the steps of cluster analysis algorithm and (iii) evaluate the quality of the established clusters. For TA, the above three problems have considered for CDF with typical articles in recent years such as [6, 7, 12] and [29]. For GA, the algorithms of [2, 13, 25, 28] and [30] have solved the problems of (i), (ii) and (iii) for CDF and CID. For CND, GA has been considered but was not exhaustive. Vovan et al. [29] proposed the automatic genetic algorithm in hard clustering for the numerical data where the problem (i) and (ii) have solved. However, the issue (iii) has not been performed. It means that the step for finding the probability to belong to clusters has not been mentioned. In addition, we see that the algorithm of [29] has disadvantage in many cases of reality because it did not improve the objective function in comparison with the traditional GA.

In this study, we propose an automatic genetic algorithm in fuzzy clustering for the numerical data (AGFC) with four modifications as follows:

(i) Propose the new objective function to optimize the result in building clusters.

(ii) Improve the crossover, mutation and selection operations of cluster analysis.

(iii) Find the probability to put in clusters of each element from the proposed objective.

(iv) Apply the proposed algorithm in recognizing the image data and building the fuzzy time series.

The proposed algorithm is performed by the established Matlab procedure. The numerical examples illustrate the proposed algorithm and test the established procedure. They show that the proposed algorithm has improved significantly the performance for CND in comparison with the existing algorithms.

The remaining content of this paper is organized as follows. Section 2 provides the objective function to build cluster using GA, and the two indexes to evaluate the quality of cluster. This section also gives the proposed algorithm. Section 3 presents the numerical examples and the convergence of the proposed algorithm. Some applications in recognizing images, building the cluster for customers and the fuzzy time series model are given in Section 4. The final section is conclusion.

2 The proposed algorithm

2.1 Some conceptions

Definition 2.1. *Let N be the number of elements in $p-$dimensions of c clusters $C_i, i = 1, 2, ..., k$. Then, the objective function of the proposed algorithm is defined as follows:*

$$FB = \frac{1}{c}\sum_{i=1}^{k} \max_{i\neq j}\left\{\frac{\frac{1}{|C_i|}\sum_{x\in C_i}\mu_{ih}(x)d(x,\bar{x}_i) + \frac{1}{|C_j|}\sum_{y\in C_j}\mu_{ih}(y)d(y,\bar{x}_j)}{d(\bar{x}_i,\bar{x}_j)}\right\}, \tag{1}$$

where x and y are the elements in groups, and $d(.)$ is the Euclidean distance of elements, $\mu_{ih}(x)$ is the fuzzy probability of h^{th} data and i^{th} cluster:

$$\mu_{ih}(x) = \frac{1}{\sum_{t=1}^{c} \left(\frac{d(x,\bar{x}_i)}{d(x,\bar{x}_t)}\right)^2}, \ 1 \le i \le c, \ 1 \le h \le N \tag{2}$$

$\bar{x}_i, \bar{x}_j$ are the prototypes of clusters C_i and C_j, respectively, $|C_i|, |C_j|$ are the number of elements in cluster C_i and C_j, respectively.

The FB index is improved from the DB index [10]. It will be used as a criterion for evaluating the quality of clustering in the proposed algorithm. The FB index computes the pairwise distances between all prototype for clusters. The minimum of these pairwise distances can be considered as the separation measurement. The more separated the clusters are, the smaller the FB index is. Also, the FB index is better than the DB index because FB is a notable representative validity index capable of handling both hard and fuzzy clusterings, while the DB index only gives the result of crisp cluster.

Definition 2.2. *The partition coefficient and entropy are used to evaluate the quality of fuzzy clustering algorithm. They are given as follows:*

$$PC = \frac{1}{N}\sum_{i=1}^{c}\sum_{h=1}^{N}\mu_{ih}^2, \tag{3}$$

$$PE = -\frac{1}{N}\sum_{i=1}^{c}\sum_{h=1}^{N}\mu_{ih}\log(\mu_{ih}), \tag{4}$$

where μ_{ih} is the fuzzy matrix belong to clusters of discrete elements calculated by (2), c and N are the number of clusters and objects, respectively. For the built algorithms, the larger of PE is, the better the algorithm is, and the PC is the opposite meaning.

2.2 The proposed algorithm

Let $Z = \{z_1, z_2, ...z_N\}$ be the considered data set, and $\varphi^{(t)} = \left\{\varphi_1^{(t)}, \varphi_2^{(t)}, \cdots, \varphi_N^{(t)}\right\}$ be the N first centroid clusters established from Z. The proposed algorithm has the following steps:

Step 1: Initiate the vector $\varphi^{(0)}$ at time $t = 0$:

$$\varphi^{(0)} = \left\{\varphi_1^{(0)}, \varphi_2^{(0)}, \ldots, \varphi_N^{(0)}\right\} = \{z_1, z_2, ..., z_N\}.$$

Step 2: Update the centroid of clusters at the time t according to equation (5):

$$\varphi_i^{(t+1)} = \frac{\sum_{j=1}^{N} \kappa\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right)\varphi_j^{(t)}}{\sum_{j=1}^{N} \kappa\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right)}, i = 1, 2, \ldots, N, \tag{5}$$

where

$$\kappa\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right) = \begin{cases} \exp(-\frac{d\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right)}{\lambda}) & \text{if } d\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right) \le \mu\alpha_{ij}(t), \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

with

$d(.)$ is the Euclidean distance, and λ is a constant,

$\alpha_{ij}(t) = \alpha_{ij}(t-1) / \left[1 + \alpha_{ij}(t-1)\,\kappa\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right)\right]$, $\alpha_{ij}(0) = 1$,

$\mu = \sum_{i<j} d\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right) / \binom{N}{2}$ is is the average of $d\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right)$.

λ is a constant, $\lambda = \frac{\sigma}{r}$, $\sigma = \sqrt{\frac{\sum_{i<j}\left[d\left(\varphi_i^{(0)}, \varphi_j^{(0)}\right) - \mu\right]^2}{\binom{N}{2}}}$ is the standard deviation, and r is a positive constant.

Step 3: Repeat Step 2 until $\max_i|\varphi_i^{(t+1)} - \varphi_i^{(t)}| < \varepsilon$.

The value of λ affects to the similar level of the elements in the established clusters because it measures the variance of the function $f_\lambda(.)$. The lower the value of λ is, the more the number of clusters for Z is. If $\lambda \to 0$, Z is divided into N clusters. In contrary, there is only one cluster if $\lambda \to \infty$. From our experiences of handling different data sets, we choose $r = 48$ or $\lambda = \sigma/48$ in this study.

Step 4: Let c be the number of the different elements in $V^{(t)}$ of Step 3. Start with c clusters presented by chromosomes for data Z. Code the chromosome by the size of cp genes representing for p-dimensional prototypes of c clusters. The genes are assigned to the non-integer values representing for the minimum/maximum value of data.

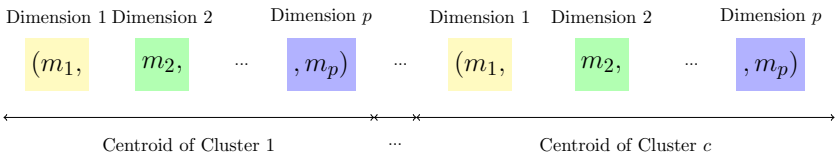


Figure 1: The method of encoding chromosomes in a population

Step 5: Initialize cp chromosomes, and evaluate their FB index using formula (1).

Step 6: Utilize the selection, crossover, and mutation operators:

- Crossover: Let L_1 and L_2 be the two parent chromosomes when the child chromosomes is created by the equation

$$Child{=}L_1 + rand * (L_2 - L_1).$$

(rand is the random vector in [0,1]).

- Let x be the value at a gene location. After mutation, x becomes x' as follows:

$$x' = x + N(0, \sigma).$$

- Selection: The main aim of this operator is select parent chromosomes for the next generation. In this paper, we use the Roulette wheel method. The probability of each individual chromosome to belong to cluster C_i is given by (7).

$$p_i = \frac{FB_i}{\sum_{j=1}^{N} FB_j}, \tag{7}$$

where FB_i is the fitness of the individual i^{th} in the population and N is the number of individuals in population.

Step 7: Compute the FB index of the chromosomes achieved in Step 6.

Step 8: Perform Step 5, Step 6, and Step 7 until the current iteration is greater than the given maximum iteration or

$$| FB_{(t)} - \overline{FB}_{(t)} | < \varepsilon,$$

where $FB_{(t)}$ is value of objective function at t^{th} iteration, and $\overline{FB}_{(t)}$ is the mean of 100 chromosomes in current population. Some parameters of genetic algorithm are given in Table 1.

The flowchart of the proposed algorithm is given by Figure 2.

3 The convergence and the numerical examples

3.1 The convergence

The proposed algorithm has two phases. Phase 1 which includes three steps (Step 1, Step 2, Step 3) determines the suitable number of clusters (DSNC). In DSNC algorithm, after each iteration, the elements in the same cluster will converge to the same value (the representative element of cluster). At the end of Step 3, if $V^{(t)}$ has c representative elements then Z will be divided to c clusters. The convergence of Phase 1 is shown by Theorem 3.1. Phase 2 is the remainder steps. This phase gives the probability to belong to the established clusters. At the end of Phase 2, the final clustering result is obtained. Phase 2 converges when the iteration reaches the maximum iteration that is proved by Theorem 3.2.

Table 1: The parameters used in the genetic algorithm

Parameter	value
Population size	100
Encoding variable	real
Chromosome length	*cp*
Generations	1000
Selection operator	Roulette
Crossover probability	0.85
Mutation probability	0.01
Maxima iterations (t)	1000
ε	0.0001

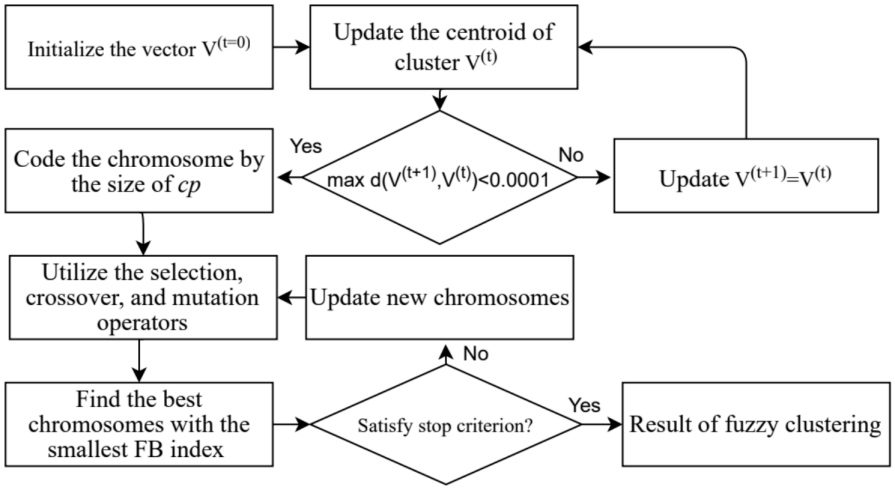


Figure 2: The flowchart of the proposed algorithm

.

Theorem 3.1. *Let $X = \{z_1, z_2, \ldots, z_N\}$ be the set of N p-dimensional elements and $\varphi^{(t)} = \left\{\varphi_1^{(t)}, \varphi_2^{(t)}, \ldots, \varphi_N^{(t)}\right\}$ be the set of N prototype elements for clusters at the t^{th} iteration. Then, the elements updated by Formulate (5) will converge to the centroid of cluster containing it.*

Proof. When $\alpha_{ij}(t) \to 0$, we have $d\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right) \leq \mu\alpha_{ij}(t) = 0$. The Euclidean distance is always greater or equal to 0, so $d\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right) = 0$ we have $\varphi_i^{(t)} = \varphi_j^{(t)}$ or

$$\lim_{\alpha(t)\to 0} \kappa\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right) = \begin{cases} 1 & \text{if } \varphi_i^{(t)} = \varphi_j^{(t)}, \\ 0 & \text{if } \varphi_i^{(t)} \neq \varphi_j^{(t)}. \end{cases} \tag{8}$$

From (5), we have $\varphi_i^{(t+1)} = \varphi_j^{(t)}$ or $\max_i d\left(\varphi_i^{(t)}, \varphi_i^{(t+1)}\right) < \varepsilon$, $\forall \varepsilon > 0$.

When $\alpha_{ij}(t) \to \infty$, we prove the elements will converge to a single cluster. Indeed, when $\alpha_{ij}(t) \to \infty$, we have $\mu\alpha_{ij}(t) \to \infty$ or $d\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right) \leq \mu\alpha_{ij}(t)$, $\forall t$ and $\lim_{\alpha(t)\to\infty} \kappa\left(\varphi_i^{(t)}, \varphi_j^{(t)}\right) = 1$. From (5) we obtain $\varphi_i^{(t+1)} = \sum_{j=1}^{N} \varphi_j^{(t)}/N$, that is, the elements converge to a single prototype. □

Theorem 3.2. *Let FB_i be the best objective function that obtained at i^{th} iterations of the genetic algorithm in Pharse 2, FB be the global value of the objective function. $\overline{FB_t} = \frac{1}{t}\sum_{i=1}^{t} FB_i$ be the average of FB_i, $\overline{FB_t^2} = \frac{1}{t}\sum_{i=1}^{t} FB_i^2$ be average of FB_i^2 at t^{th} iterations. Then, Phase 2 of the proposed algorithm converges because*

$$\lim_{t\to\infty} P\left(\mid FB_t - \overline{FB_t} \mid \leq \varepsilon\right) = 1.$$

Proof. Let $\varepsilon_0 > 0$, we have $\lim_{t\to\infty} P\left(\mid FB_t - FB \mid > \varepsilon_0\right) = 0$.

On the other hands, Let $\varepsilon_1 > 0$, exist $t_1 < t$ so that

$$1 - P\left(\mid FB_t - FB \mid \leq \varepsilon_0\right) < \varepsilon_1 \Rightarrow P\left(\mid FB_t - FB \mid \leq \varepsilon_0\right) > 1 - \varepsilon_1. \tag{9}$$

Let $\varepsilon > 0$ and $t \to \infty$. Then, $\lim_{t\to\infty} P\left(\frac{1}{t}\sum_{i=1}^{t}\left(FB_i - \overline{FB_t}\right)^2 \leq \varepsilon\right) = 1$.

We have

$$\begin{aligned}
\frac{1}{t}\sum_{i=1}^{t}\left(FB_i - \overline{FB_t}\right)^2 &= \frac{1}{t}\sum_{i=1}^{t}\left[(FB_i - FB) - \left(\overline{FB_t} - FB\right)\right]^2 \\
&= \frac{1}{t}\sum_{i=1}^{t}(FB_i - FB)^2 - \left(\overline{FB_t} - FB\right)^2 \\
&\leq \frac{1}{t}\sum_{i=1}^{t}(FB_i - FB)^2.
\end{aligned} \tag{10}$$

If $t_0 < t$, then

$$\frac{1}{t}\sum_{i=1}^{t}(FB_i - FB)^2 = \frac{1}{t}\sum_{i=1}^{t_0}(FB_i - FB)^2 + \frac{1}{t}\sum_{i=t_0+1}^{t}(FB_i - FB)^2.$$

Because FB_1 is the best objective function and $FB \leq FB_i \leq FB_1$,
Therefore,

$$\frac{1}{t}\sum_{i=1}^{t_0}(FB_i - FB)^2 \leq \frac{1}{t}\sum_{i=1}^{t_0}(FB_1 - FB)^2 = \frac{t_0}{t}(FB_1 - FB)^2.$$

In other words, there exist $t_1 > t_0$ and $\varepsilon_2 > 0$ so that $\frac{t_0}{t_1}(FB_1 - FB)^2 < \varepsilon_2$.

If $t > t_1 > t_0$, then

$$\frac{1}{t}\sum_{i=1}^{t_0}(FB_i - FB)^2 \leq \frac{t_0}{t}(FB_1 - FB)^2 \leq \frac{t_0}{t_1}(FB_1 - FB)^2 \leq \varepsilon_2.$$

Because $FB_1 \geq FB_2 \geq ... \geq FB_i \geq FB_{i+1} \geq ... \geq FB$, so

$$\begin{aligned}
\frac{1}{t}\sum_{i=t_0+1}^{t}(FB_i - FB)^2 &\leq \frac{1}{t}\sum_{i=t_0+1}^{t}(FB_{t_0+1} - FB)^2 \\
&= \frac{t - t_0 - 1}{t}(FB_{t_0+1} - FB)^2 \\
&\leq (FB_{t_0+1} - FB).
\end{aligned}$$

From equation (9), we have $P\left(\mid FB_t - FB \mid \leq \varepsilon_0\right) > 1 - \varepsilon_1$.

If $t > t_0$, then

$$P\left((FB_{t_0+1} - FB)^2 \leq \varepsilon_0^2\right) > 1 - \varepsilon_1 \Rightarrow P\left((FB_{t_0+1} - FB)^2 \leq \varepsilon_0\right) > 1 - \varepsilon_1, \varepsilon_0 \ll 1.$$

Let $\varepsilon = \varepsilon_1 + \varepsilon_2$, we have

$$\begin{aligned}
P\left(\frac{1}{t}\sum_{i=1}^{t}(FB_i - FB)^2 \leq \varepsilon\right) &= P\left(\frac{1}{t}\sum_{i=1}^{t_0}(FB_i - FB)^2 + \frac{1}{t}\sum_{i=t_0+1}^{t}(FB_i - FB)^2 \leq \varepsilon\right) \\
&\geq P\left(\varepsilon_2 + \frac{1}{t}\sum_{i=t_0+1}^{t}(FB_i - FB)^2 \leq \varepsilon\right) \\
&= P\left(\frac{1}{t}\sum_{i=t_0+1}^{t}(FB_i - FB)^2 \leq \varepsilon - \varepsilon_2\right) > 1 - \varepsilon_1.
\end{aligned}$$

Therefore, we can conclude that $\varepsilon_1 > 0$, exist $t_1 < t$ so that

$$P\left(\frac{1}{t}\sum_{i=1}^{t}(FB_i - FB)^2 \leq \varepsilon\right) > 1 - \varepsilon_0.$$

or

$$\lim_{t\to\infty} P\left(\frac{1}{t}\sum_{i=1}^{t}(FB_i - FB)^2 \leq \varepsilon\right) = 1.$$

From (10), we have $\lim\limits_{t\to\infty} P\left(\frac{1}{t}\sum\limits_{i=1}^{t}\left(FB_i - \overline{FB_t}\right)^2 \leq \varepsilon\right) = 1$. According to Chebyshev's Theorem, Theorem 3.2 has been proved. □

3.2 The numerical examples

Example 3.3. *In this example, we use 200 elements simulated from three Gaussian distributions with means and covariance matrices as follows:*

$$\text{Cluster } 1: \mu_1 = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}.$$

$$\text{Cluster } 2: \mu_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 0.1 & 0.05 \\ 0.05 & 0.1 \end{pmatrix}.$$

$$\text{Cluster } 3: \mu_3 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \Sigma_3 = \begin{pmatrix} 0.1 & -0.05 \\ -0.05 & 0.1 \end{pmatrix}.$$

$$\text{Cluster } 4: \mu_4 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma_4 = \begin{pmatrix} 0.1 & -0.01 \\ -0.01 & 0.1 \end{pmatrix}.$$

Extract two-dimensional 50 points from each cluster, we have the actual result of clusters:

$$C_1 = \{z_1, z_2, ..., z_{50}\}; \quad C_2 = \{z_{51}, z_{52}, ..., z_{100}\};$$
$$C_3 = \{z_{101}, z_{102}, ..., z_{150}\}; \quad C_4 = \{z_{151}, z_{152}, ..., z_{200}\}.$$

where

$$z_1 = (0.52, 0.12), z_2 = (-0.195, 0.468), ..., z_{199} = (1.817, 2.172), z_{200} = (1.798, 2.381).$$

The scatter of this dataset is shown in Figure 2.

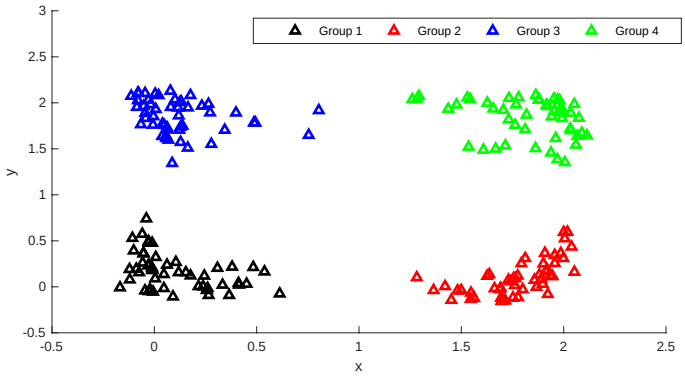


Figure 3: The scatter plot shows that the 200 points extracted from three distribution functions

Using Phase 1 to determine the suitable number of clusters, we have:

Step 1: *Initiate* $\varphi^{(0)}$.

$$\varphi^{(0)} = \{(0.52, 0.12), (-0.195, 0.468), ..., (1.817, 2.172), (1.798, 2.381)\}.$$

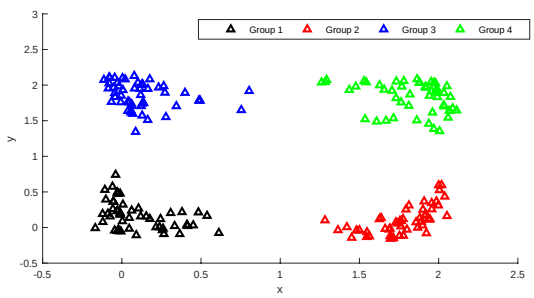
Step 2: *Update the centroid at* $t = 1$, *we have*

$$\varphi^{(1)} = \{(0.538, 0.164), (-0.029, 0.493), ..., (1.767, 1.98), (1.732, 2.05)\},$$

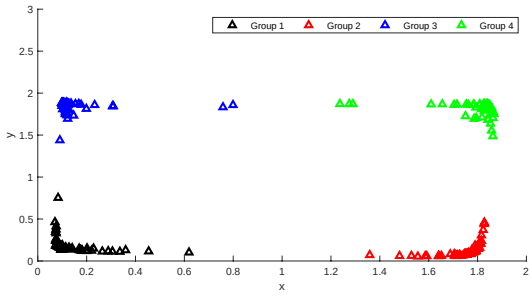
where

$$\kappa\left(\varphi_i^{(0)}, \varphi_j^{(0)}\right) = \begin{bmatrix} 1 & 0.423 & \cdots & 0 & 0 \\ 0.423 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0.798 \\ 0 & 0 & \cdots & 0.798 & 1 \end{bmatrix}, \ \mu = 1.898, \ \sigma = 0.924.$$

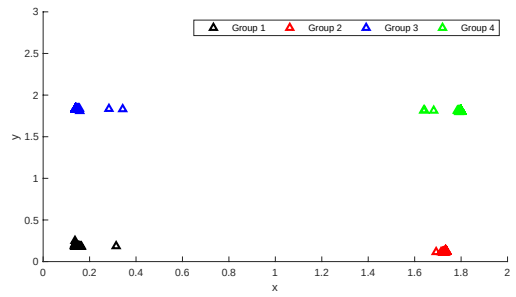
Step 3: *Because* $\max_i|\varphi_i^{(t+1)} - \varphi_i^{(t)}| = 0.822 > \varepsilon = 0.0001$, *the algorithm will repeat Step 2 until the stopped criterion is satisfied. After five iterations, the algorithm obtains the suitable number of clusters shown by Figure 4.*



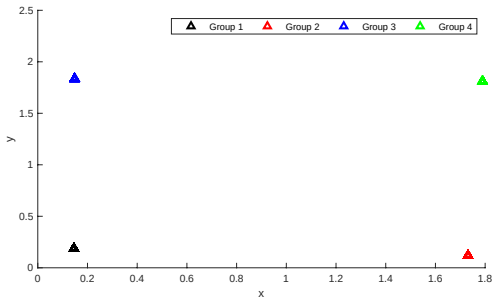
((a)) Iteration 1



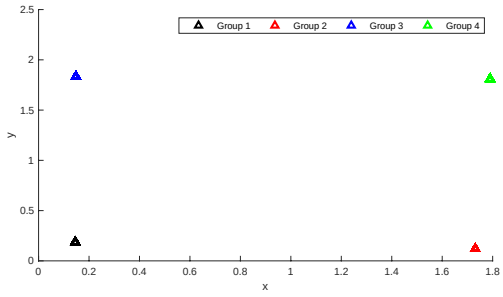
((b)) Iteration 2



((c)) Iteration 3



((d)) Iteration 4



((e)) Iteration 5

Figure 4: The convergence of Phase 1 through the five iterations

We can see that 200 points converge into the four clusters. In detail, after each iteration, the elements concentrate on its nearest center. The final iteration obtains 4 elements. It means that we have 4 clusters.
Run Phase 2 of the algorithm, we have:

Steps 4 and 5: *Code the chromosome with the size as 8, and create 100 chromosomes in populations, we have*

$$m = [-0.008, 0.067, 1.795, -0.213, 1.862, 2.115, -0.185, 2.46] .$$

Step 6 and Step 7: *Perform the operators of genetic algorithm, and calculate the best FB index of the new population, we have $FB = 0.523$.*
Step 8: *Calculate*

$$|FB^{(1)} - \overline{FB^{(1)}}| = |0.523 - 2.7| = 2.177 > 0.0001.$$

Repeat 1000 iterations, we have the result as follows:

- *The optimal cluster:*

$$C_1 = \{z_1, z_2, ..., z_{50}\}; C_2 = \{z_{51}, z_{52}, ..., z_{100}\};$$
$$C_3 = \{z_{101}, z_{102}, ..., z_{150}\}; C_4 = \{z_{151}, z_{152}, ..., z_{200}\}.$$

- *The value of objective function:* $FB = 0.382$.

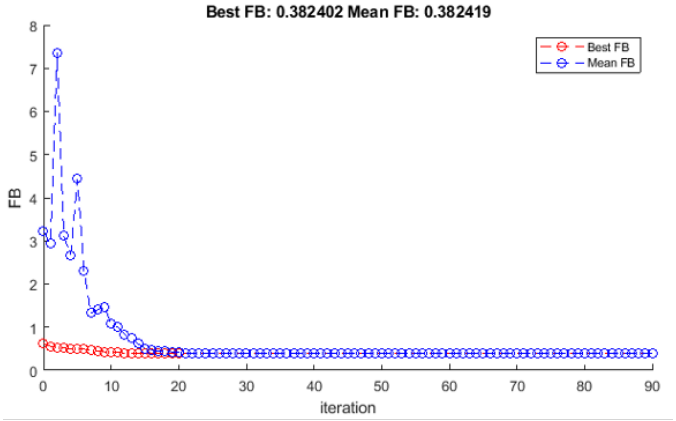


Figure 5: The convergence of Phase 2 for 200 elements

The fuzzy membership between each element and the established clusters is shown by Figure 6.

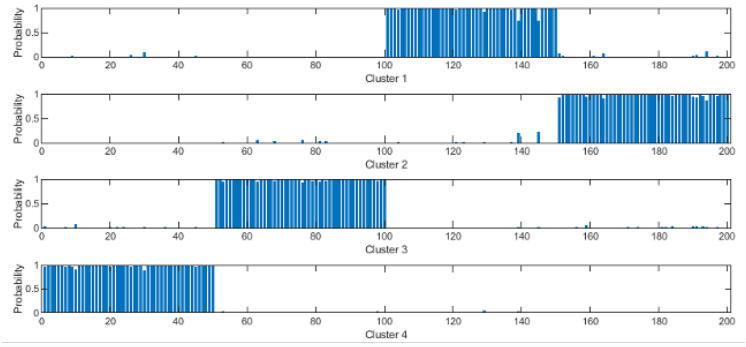


Figure 6: The fuzzy membership of 200 elements and the established clusters

Figure 6 shows that the proposed algorithm has the accuracy as the real outcome. In addition, it also determines the good probability of the element to belong to each cluster. Performing the proposed and well-known algorithms for 50 times, we obtain the average value of PE and PC in Table 2.

According to Table 2, it can be seen that the proposed algorithm has the best outcome with the largest value of PC index and the smallest of PE index.

In order to evaluate the statistical significance for the theses differences, we use the analysis of variance (ANOVA). For the PE index, the result of test is shown in Table 7.

Table 2: The average value of PE and PC for algorithms

Algorithm	PE	PC
FCM	0.306	0.867
Tai and Nguyen Trang [28]	0.128	0.898
Proposed	0.068	0.973

Table 3: ANOVA for the PE index of the methods

Source of Variation	SS	df	MS	F	P-value
Between Groups	1.56	2	0.780	11471.52	0.000
Within Groups	0.01	147	0.00007		
Total	1.57	149			

Table 3 shows that it has a significant difference in the average PE index of the algorithms. The Post hoc with Tukey test also shows that there are the significant difference between the proposed algorithm with the algorithms of [28] *and FCM.*

A similar process is also performed for the PC index to conclude that the proposed algorithm gives the best result, and there is the significant difference about the PC index between the proposed algorithm with the considered approaches.

Example 3.4. *We consider three benchmark data sets: Iris flower, Wine, and Liver disorder represented in many studies such as* [14] *and* [3]. *The general information about theses data is given in Table 4.*

Table 4: The general information of 3 data sets

Dataset	Size	Dimension	No. of clusters
Iris	150	4	3
Wine	178	13	3
Liver disorder	345	7	2

The purpose of this example is to compare the effectiveness of the proposed algorithm with others. Comparing the proposed algorithm and others through out the average of PE and PC indexes with 50 times of performing, we obtain Table 5.

Table 5: The result of algorithms for 3 benchmark data sets

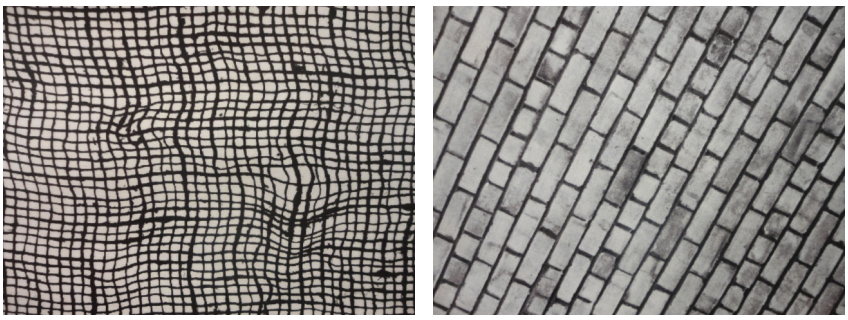
Dataset	Method	PE	PC
Iris	FCM	0.395	0.783
	Tai and Nguyen Trang [28]	0.237	0.815
	Proposed	0.143	0.919
Wine	FCM	0.380	0.791
	Tai and Nguyen Trang [28]	0.259	0.826
	Proposed	0.916	0.141
Liver disorder	FCM	0.288	0.830
	Tai and Nguyen Trang [28]	0.127	0.879
	Proposed	0.049	0.972

For all data, Table 5 shows that the PC index of the proposed algorithm is highest while its PE is lowest. The value of these indexes are quite far from the other algorithms. The a one-way ANOVA analysis analysis and Post hoc with Turkey test also show that there are significance differences between the proposed algorithm and the considered approaches.

4 Some applications

4.1 Image recognition

This application utilizes the proposed algorithm for image recognition. The algorithm is tested on Brodatz's data, which is divided into two groups, each containing 21 images. The dataset used for this purpose is provided by the source http://imagem.sel.eesc.usp.br/base/Brodatz-rotated/index.html. Figure 7 shows some sample images from the two groups.



((a)) Group 1 ((b)) Group 2

Figure 7: The two sample images of two groups

For each image, we extracted data using the Gray Level Co-occurrence matrix (GLCM), which yielded four features: Contrast, Correlation, Energy, and Homogeneity. The theoretical aspects of image extraction can be found in relevant research studies such as [23, 29]. The results of the extraction process are presented in Table 6 and depicted in Figure 8.

Table 6: The result of extracting the four features of images

Image	Contrast	Correlation	Energy	Homogeneity
I_1	0.808	0.908	0.113	0.807
I_2	1.036	0.887	0.129	0.761
...	...	...	...	...
I_{42}	0.330	0.965	0.151	0.881

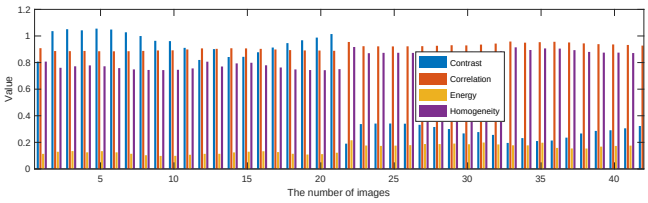


Figure 8: The value of four features of the images

After extracting the features from the images, we proceed to determine the optimal number of groups for this dataset. The results of Phase 1 are displayed in Figure 9.

From Figure 9, we can see that each feature converges on two value clearly. For example, the Contrast feature in 21 of first images has the same values as 0.956 and 0.280 for the remaining images. Ending this phase, we have two clusters. Continue to perform Phase 2 with 80 iterations, we have the result shown by Figure 10.

Next, we calculate the probabilities assigned to image clusters, as illustrated in Figure 11.

When comparing the proposed algorithm to other algorithms, we refer to Table 7.

Table 7 presents the outstanding results achieved by the proposed algorithm in comparison to existing algorithms, as indicated by both the PE and PC indexes.

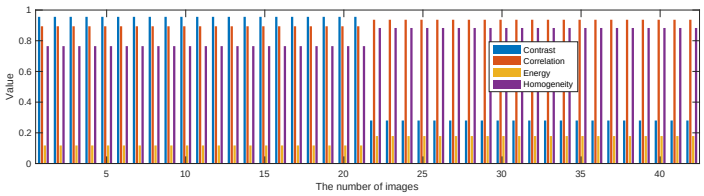


Figure 9: The value of four features of the images in the final iteration

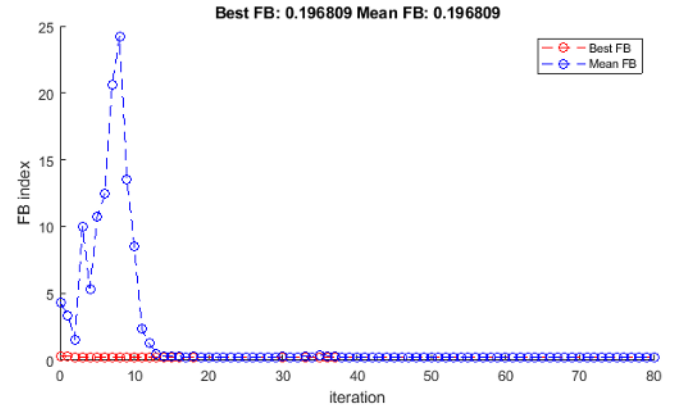


Figure 10: The convergence of the Phase 2 for images

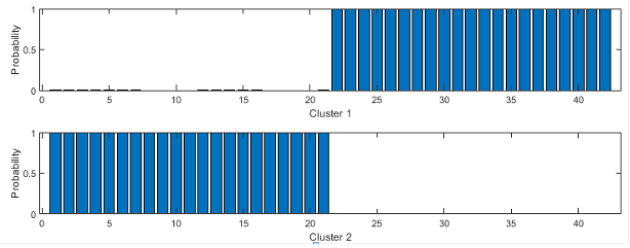


Figure 11: The fuzzy membership of 42 elements and the established clusters

.

Table 7: The result of the proposed algorithm and others for image data

Method	PE	PC
FCM	0.306	0.867
Tai and Nguyen Trang [28]	0.265	0.907
Proposed	0.068	0.973

4.2 Clustering for customers

This dataset is specifically designed for learning purposes and focuses on customer segmentation, also known as market basket analysis. The dataset contains information such as CustomerID, Gender, Age, Annual income ($), and spending score and presented in Table 8. It is sourced from the open-source: www.kaggle.com vjchoudhary7/customer-segmentation-tutorial-in-python.

By performing 7 iterations of Phase 1, we determine the optimal number of clusters for the customers. Figure 12 illustrates the results obtained from this analysis.

Figure 12 illustrates that when starting with 200 initial elements, they converge to 7 representative elements. This implies that we end up with 7 clusters.

Afterward, we proceed with Phase 2 of the proposed algorithm, which is illustrated by the process depicted in Figure 13.

Table 8: Information of customer dataset

Variables	Value
Size	200
Age	18-70
Gender	40% of Male and 60% of Female
Annual Income	15-137 (thousand USD)
Spending Score	1-100

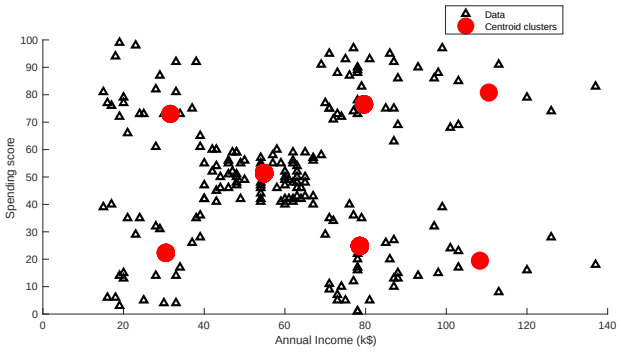


Figure 12: The Phase 1 of customer data set

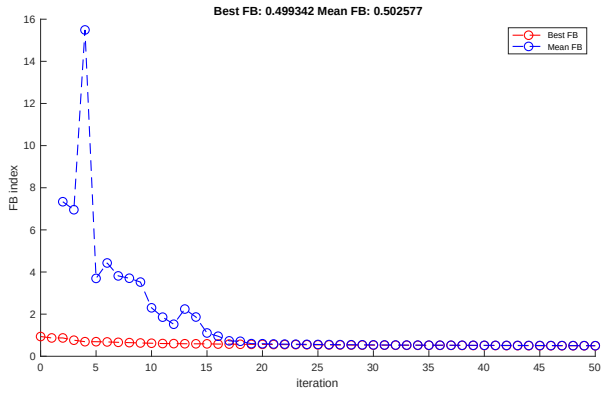


Figure 13: The convergence for 200 customers

At the end of this phase, we obtain the probabilities of belonging to the 7 clusters, as demonstrated in Figure 14. Compare the result of algorithms, we have Table 9

Table 9: Comparing the proposed algorithm and others for customer data set

Method	PE	PC
FCM	0.867	0.606
Tai and Nguyen Trang [28]	0.519	0.785
Proposed	0.309	0.864

Table 9 once again validates that the proposed algorithm yields the best results.

In summary, across various examples and applications encompassing differences in data types, such as numbers, characters, and fields, the proposed algorithm consistently delivers exceptional performance when compared to other approaches.

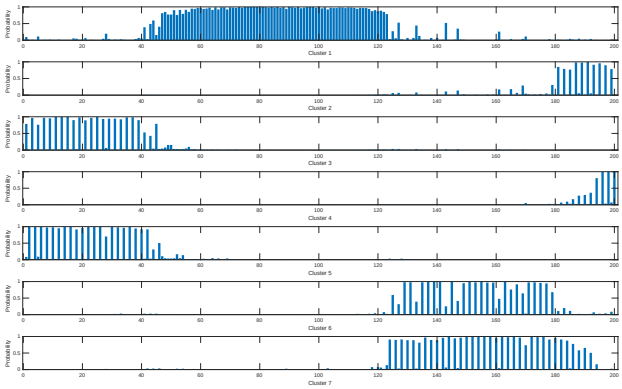


Figure 14: The fuzzy numbers of 200 customers belong to 7 groups

4.3 Building the time series model

This section presents the application of the proposed algorithm to build the fuzzy time series model. The proposed model consists of the following steps for a given time series, denoted as $X = \{x_1, x_2, ..., x_n\}$:

Step 1: Apply Phase 1 of the proposed algorithm to divide the series X into clusters with a suitable number. Let's assume that after this step, we have h clusters.

Step 2: Establish the initial partition matrix $V^{(0)} = [\mu_{ij}^{(0)}]_{h\times n}$, where $\mu_{ij}^{(0)}$ represents the probability of assigning the jth element of the series to the ith cluster, as determined in Step 1.

Step 3: Perform the steps of Phase 2 in the proposed algorithm to update the initial partition matrix $V^{(0)}$ until convergence. This iterative process yields the final partition matrix $V^{(m)} = [\mu_{ij}^{(m)}]_{h\times n}$.

Step 4: Forecast the series using the following principle:

$$x_j = \mu_{ij}^{(m)}.V_i, i = 1, 2, ..., h; j = 1, 2, ..., n. \tag{11}$$

where V_i is the centroid of the ith cluster.

For the proposed time series model, we apply it to forecast the peak salinity levels at two stations located on the main rivers in Tra Vinh province, a coastal province in Vietnam. Vietnam, as you may know, is one of the countries heavily affected by climate change. Among the various impacts of climate change in Vietnam, salty intrusion in the coastal provinces is considered one of the most severe. It has caused significant damage to agriculture and livestock in recent years. To address this problem, there is an urgent need for scientists to predict the future levels of salty intrusion. The forecasting results will serve as a crucial scientific basis for implementing appropriate measures and minimizing the impacts.

Despite many efforts being made, this problem remains unresolved to date. In this study, we utilize the data sets from Travinh1 and Travinh2 stations, as presented in Table 10.

By applying the proposed model, we have generated the "Forecasting" column in Table 10, which is also depicted in Figure 15 and Figure 16.

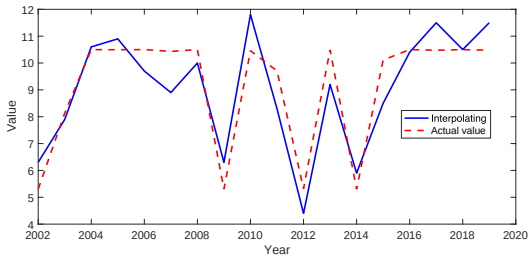


Figure 15: The outcome of the real and forecasting data for Travinh1

Table 10: The result of training data

Year	Travinh1	Forecasting	Travinh2	Forecasting
2002	6.3	5.4	7.9	6.54
2003	7.9	8.0	11.3	12.3
2004	10.6	10.4	8.3	7.89
2005	10.9	10.3	10.7	11.2
2006	9.7	10.4	9	9.49
2007	8.9	10.0	9.5	9.51
2008	10	10.5	9.9	9.51
2009	6.3	5.4	9.9	9.51
2010	11.8	10.0	10.8	11.7
2011	8.3	8.9	11.1	12.3
2012	4.4	5.6	9.1	9.5
2013	9.2	10.2	12.4	12.3
2014	5.9	5.3	6	6.3
2015	8.5	9.4	8.9	9.45
2016	10.4	10.4	10.7	11.2
2017	11.5	10.1	11.7	12.3
2018	10.5	10.4	11.4	12.3
2019	11.5	10.1	11.9	12.3

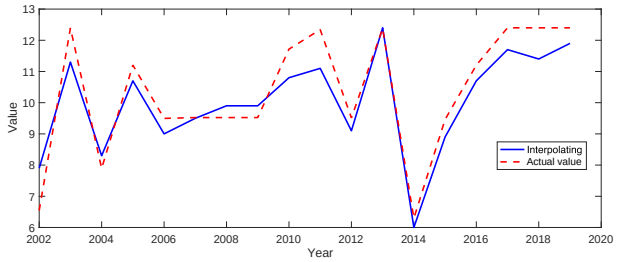


Figure 16: The outcome of the real and forecasting data for Travinh2

By comparing the proposed model with other models such as ARIMA, LSTM, Abbasov and Mamedova [1], Vovan [26], Vovan et al. [27], using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), we have compiled the results in Table 11.

Table 11: The parameter results of considered methods for two datasets

Dataset	Model	MAE	MAPE	MSE
Travinh1	Proposed	0.83	10.04	0.93
	ARIMA	2.91	38.54	14.78
	Abbasov and Mamedova [1]	4.36	62.29	19.10
	LSTM	1.39	12.73	2.43
	Vovan [26]	1.18	11.98	1.97
	Vovan et al. [27]	1.80	17.91	3.30
Travinh2	Proposed	0.60	6.06	0.50
	ARIMA	1.02	10.75	1.55
	Abbasov and Mamedova [1]	3.13	35.62	12.61
	LSTM	2.19	18.78	4.87
	Vovan [26]	1.27	11.02	1.72
	Vovan et al.[27]	0.70	6.22	0.74

Table 11 clearly demonstrates that the proposed approach outperforms the other models in terms of all the evaluated parameters, including MAE, MAPE, and MSE, for both data sets.

5 Conclusion

In this study, we have developed a new fuzzy cluster analysis algorithm specifically designed for discrete elements. This algorithm incorporates various improvements, including enhancements to the objective function of the genetic algorithm, a method to determine the optimal number of clusters, and a step to calculate the probability of each element belonging to specific clusters. These enhancements have significantly improved the quality of clustering. With the implementation of the proposed algorithm using a Matlab procedure, it becomes effortless to apply it to real-world data. Through extensive testing on multiple datasets, the proposed algorithm consistently produced stable and reasonable results, surpassing the performance of existing algorithms. Moreover, the algorithm's contributions extend beyond clustering and find applications in data analysis, image recognition, and fuzzy time series modeling. These practical applications demonstrate the algorithm's potential and relevance in various domains. This research can be extended for practical applications in other fields such as security, medical, and economics. Furthermore, these applications also serve as our future research directions.

Acknowledgements

This study was financially supported by Van Lang University, Vietnam, under grant number 13/2022/HD-NCKH

References

[1] A. M. Abbasov, M. B. Mamedova, *Application of fuzzy time series to population forecasting*, Vienna University of Technology, **12** (2003), 545-552.

[2] L. Agusti, S. Salcedo Sanz, S. Jiménez-Fernández, L. Carro Calvo, J. Del Ser, J. A. Portilla-Figueras, *A new grouping genetic algorithm for clustering problems*, Expert Systems with Applications, **39**(10) (2012), 9695-9703.

[3] A. Asuncion, D. Newman, *Uci machine learning repository*, University of California, 2007.

[4] P. Berkhin, *A survey of clustering data mining techniques*, In: Kogan, J., Nicholas, C., Teboulle, M. (eds) Grouping Multidimensional Data. Springer, Berlin, 2006.

[5] J. C. Bezdek, R. Ehrlich, W. Full, *FCM The fuzzy c-means clustering algorithm*, Computers and Geosciences, **10**(2-3) (1984), 191-203.

[6] N. Bidi, Z. Elberrichi, *Feature selection for text classification using genetic algorithms*, In 2016 8th International Conference on Modelling, Identification and Control, IEEE, (2016), 806-810.

[7] N. Bouguila, W. ElGuebaly, *Discrete data clustering using finite mixture models*, Pattern Recognition, **42**(1) (2009), 33-42.

[8] J. H. Chen, W. L. Hung, *An automatic clustering algorithm for probability density functions*, Journal of Statistical Computation and Simulation, **85**(15) (2015), 3047-3063.

[9] M. Chen, D. Miao, *Interval set clustering*, Expert Systems with Applications, **38**(4) (2011), 2923-2932.

[10] D. L. Davies, D. W. Bouldin, *A cluster separation measure*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **2** (1979), 224-227.

[11] F. D. A. De Carvalho, J. T. Pimentel, L. X. Bezerra, *Clustering of symbolic interval data based on a single adaptive* L^1 *distance*, In 2007 International Joint Conference on Neural Networks, IEEE, (2007), 224-229.

[12] E. Egrioglu, C. H. Aladag, U. Yolcu, *Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks*, Expert Systems with Applications, **40**(3) (2013), 854-857.

[13] A. Goh, R. Vidal, *Unsupervised riemannian clustering of probability density functions*, In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008, Antwerp, Belgium, (2008), 377-392.

[14] L. O. Hall, I. B. Ozyurt, J. C. Bezdek, *Clustering with a genetically optimized approach*, IEEE Transactions on Evolutionary Computation, **3**(2) (1999), 103-112.

[15] W. L. Hung, J. H. Yang, K. F. Shen, *Self-updating clustering algorithm for interval-valued data*, In 2016 IEEE International Conference on Fuzzy Systems, (2016), 1494-1500.

[16] J. T. Jeng, C. M. Chen, S. C. Chang, C. C. Chuang, *Ipfcm clustering algorithm under euclidean and Hausdorff distance measure for symbolic interval data*, International Journal of Fuzzy Systems, **21**(7) (2019), 2102-2119.

[17] H. Le Capitaine, C. Frelicot, *A cluster-validity index combining an overlap measure and a separation measure based on fuzzy-aggregation operators*, IEEE Transactions on Fuzzy Systems, **19**(3) (2011), 580-588.

[18] T. W. Liao, *Clustering of time series data a survey*, Pattern Ecognition, **38**(11) (2005), 1857-1874.

[19] L. Ma, Y. Zhang, V. Leiva, S. Liu, T. Ma, *A new clustering algorithm based on a radar scanning strategy with applications to machine learning data*, Expert Systems with Applications, **191** (2022), 116143.

[20] A. Montanari, D. G. Calò, *Model-based clustering of probability density functions*, Advances in Data Analysis and Classification, **7**(3) (2013), 301-319.

[21] H. Nguyen, X. N. Bui, Q. H. Tran, N. L. Mai, *A new soft computing model for estimating and controlling blast-produced ground vibration based on hierarchical k-means clustering and cubist algorithms*, Applied Soft Computing, **77** (2019), 376-386.

[22] T. Nguyentrang, T. Vovan, *Fuzzy clustering of probability density functions*, Journal of Applied Statistics, **44**(4) (2017), 583-601.

[23] D. Phamtoan, T. Vovan, *Automatic fuzzy genetic algorithm in clustering for images based on the extracted intervals*, Multimedia Tools and Applications, **80**(28) (2021), 35193-35215.

[24] S. I. R. Rodríguez, F. D. A. T. De Carvalho, *A new fuzzy clustering algorithm for interval-valued data based on city-block distance*, IEEE International Conference on Fuzzy Systems, (2019), 1-6.

[25] M. Rostami, P. Moradi, *A clustering based genetic algorithm for feature selection*, In 2014 6th Conference on Information and Knowledge Technology, (2014), 112-116.

[26] T. Vovan, *An improved fuzzy time series forecasting model using variations of data*, Fuzzy Optimization and Decision Making, **18**(2) (2019), 151-173.

[27] T. Vovan, L. Nguyenhuynh, K. Nguyenhuu, *Building the forecasting model for time series based on the improvement of fuzzy relationships*, Iranian Journal of Fuzzy Systems, **19**(4) (2022), 89-106.

[28] T. Vovan, T. Nguyentrang, *Similar coefficient of cluster for discrete elements*, Sankhya B, **80**(1) (2018), 19-36.

[29] T. Vovan, D. Phamtoan, D Tranthituy, *Automatic genetic algorithm in clustering for discrete elements*, Communications in Statistics-Simulation and Computation, **50**(6) (2021), 1679-1694.

[30] T. Vovan, D. Phamtoan, L. H. Tuan, T. Nguyentrang, *An automatic clustering for interval data using the genetic algorithm*, Annals of Operations Research, **303**(1) (2021), 359-380.

[31] Q. Wang, X. Wang, C. Fang, W. Yang, *Robust fuzzy c-means clustering algorithm with adaptive spatial and intensity constraint and membership linking for noise image segmentation*, Applied Soft Computing, **92** (2020), 106318.

[32] K. L. Wu, M. S. Yang, *A cluster validity index for fuzzy clustering*, Pattern Recognition Letters, **26**(9) (2005), 1275-1291.

[33] H. Yu, L. Chen, J. Yao, X. Wang, *A three-way clustering method based on an improved dbscan algorithm*, Physica A: Statistical Mechanics and its Applications, **535** (2019), 122289.

[34] X. Zhao, J. Liang, C. Dang, *A stratified sampling based clustering algorithm for large-scale data*, Knowledge-Based Systems, **163** (2019), 416-428.