



# Building fuzzy time series model from unsupervised learning technique and genetic algorithm

Dinh Phamtoan<sup>1,2,3</sup> · Tai Vovan<sup>4</sup>

Received: 22 August 2020 / Accepted: 30 August 2021 / Published online: 18 October 2021  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

## Abstract

This paper proposes a new model to interpolate time series and forecast it effectively for the future. The important contribution of this study is the combination of optimal techniques for fuzzy clustering problem using genetic algorithm and forecasting model for fuzzy time series. Firstly, the proposed model finds the suitable number of clusters for a series and optimizes the clustering problem by the genetic algorithm using the improved Davies and Bouldin index as the objective function. Secondly, the study gives the method to establish the fuzzy relationship of each element to the established clusters. Finally, the developed model establishes the rule to forecast for the future. The steps of the proposed model are presented clearly and illustrated by the numerical example. Furthermore, it has been realized positively by the established MATLAB procedure. Performing for a lot of series (3007 series) with the differences about characteristics and areas, the new model has shown the significant performance in comparison with the existing models via some parameters to evaluate the built model. In addition, we also present an application of the proposed model in forecasting the COVID-19 victims in Vietnam that it can perform similarly for other countries. The numerical examples and application show potential in the forecasting area of this research.

**Keywords** Cluster analysis · Forecast · Fuzzy time series · Interpolate

## 1 Introduction

We all agree that forecasting is the scientific basis for the good plans required for many areas. Because of its important role in many fields, forecasting always gets the attention of managers and scientists. Despite several discussions in the literature, the problems of forecasting have not yet been completely solved [1, 21]. In statistics, time series and regression are popular models applied to

forecast, but they have many disadvantages in practice. When building a regression model, we must constrain the data conditions that do not satisfy for real data. Therefore, it often receives the limited results in forecasting [2, 5, 22, 35].

In socioeconomic development, each field and country has stored a lot of data over time. Therefore, time series has become the most common data type. For these data, forecasting is the most attractive direction. There are two kinds in building the time series. They are non-fuzzy time series (NFS) and fuzzy time series (FS) models. Although NFS models often have more advantages than regression models in the real application, they have some limitations. For example, they only give the remarkable results if the series have normal changes or stationary [46]. Based on historical data, the NFS sets up a mathematical function to forecast, so it had not much flexibility. Non-reliance on the linguistic level to build the relationship of the elements in series is considered the main limitation of the NFS models. Because the FS models are built based on the fuzzy

---

✉ Tai Vovan  
vvtai@ctu.edu.vn

Dinh Phamtoan  
dinh.pt@vlu.edu.vn

<sup>1</sup> University of Science, Ho Chi Minh City, Vietnam

<sup>2</sup> Vietnam National University, Ho Chi Minh City, Vietnam

<sup>3</sup> Faculty of Engineering, Van Lang University,  
Ho Chi Minh City, Vietnam

<sup>4</sup> College of Natural Science, Can Tho University,  
Can Tho City, Vietnam

relations of the elements in series, they have overcome the weaknesses of NFS models.

The FS model is developed in two main directions. First, it builds the models from the original series and forecasts for the future from these models themselves. Abbasov and Manedova [1] and Tai et al. [46] have had the important contributions in this direction. Second, the original series are interpolated in order to obtain the new ones that are closely related to each other in the whole series. After that, this new series is used as the good input data to forecast. Compared to the first direction, the second one is getting more attention in our knowledge. Song et al. [42] were the pioneer in this direction with data on the enrollment of the University of Alabama (EnrollmentUA). Qiang et al. [43] used the triangular fuzzy relation for performing. Ming et al. [14] and Chen et al. [16] improved the result of [43] when taking notice of fuzzy level. Huarng [27] and Own [37] presented a heuristic model for FS using heuristic knowledge to improve the forecast for EnrollmentUA. Based on neural network, Alpaslan [7] gave the interesting results in some cases. Wu and Chau [50] constructed several soft computing approaches for rainfall prediction. Two aspects were considered improving the accuracy of rainfall prediction: carrying out a data preprocessing procedure and adopting a modular method. The proposed techniques included the moving average (MA) and singular spectrum analysis (SSA). The modular models were composed of local support vectors regression (SVR) model and local artificial neural networks (ANNs) model. Results showed that the MA was superior to the SSA when they were coupled with the ANN. Riccardo and Kwok [48] proposed the artificial neural networks-based interval forecasting of streamflow discharges using the lower and upper bounds and multi-objective fully informed particle swarm. Ghandari et al. [24] introduced the aeromechanical optimization technique of first row compressor test stand blades using a hybrid machine learning model of genetic algorithm and artificial neural network. The authors used three-dimensional geometric parameters to conduct blade tuning. As a result, the reduced frequency increases by at least 5% in both stall and classical regions, and force response constraints are satisfied.

From the fuzzy model in accordance with different linguistic levels, many scientists such as [25, 34, 49] have proposed the new models. Moreover, Baghban et al. [10] used the adaptive network-based fuzzy inference system (ANFIS) which provided highly accurate predictions. The study was expanded based on the independent variables of temperature, nanoparticle diameter, nanofluid density, volumetric fraction, and viscosity of the base fluid. Prashant et al. [38] presented a fuzzy dominance-based

analytical sorting method as advancement to the existing multi-objective evolutionary algorithm. The objective functions are defined as fuzzy objectives, and competing solutions are provided an overall activation score based on their respective fuzzy objective values. Recently, Tai [47] proposed a FS model from the results of the fuzzy clustering problem. Many applications have used the optimal techniques such as bat algorithm [51], genetic algorithm [38], and whale algorithm [36] in recent years. In this study, we consider the genetic algorithm to apply to FS model. Applying the genetic algorithm (GA) in clustering, Jain [30] proposed the FS for EnrollmentUA. Ali et al. [6] proposed a method based on a genetic algorithm (GA) for generation expansion planning (GEP) in the presence of wind power plants. A six-state model was used to obtain the wind farm output power model. The method of calculating the six-state wind farm output model with the turbine's forced outage rate of wind farm units for use in long-term GEP calculations is described. Also, using GA, Aldouri et al. [3] introduced a model with two levels. The first level implements GA based on the autoregressive integrated moving average (ARIMA) model. The second level is utilized based on the forecasting error rate.

In time series, each value on time  $t$  is called the element, and the universal set is a set which contains all the elements as input data to forecast. For the second direction, a time series model is built to have three main phases. (1) Build the universal set and divide the suitable groups for it, (2) determine the elements for each group, and (3) establish the relationship between each element of the series to the groups found from (2) to build the rule for forecasting. For (1), many authors used the universal set to be the original series itself [13, 16, 23]. Some others used the maximum value and minimum value of original series to make the universal set [13, 16]. In addition, Huarng et al. [27, 28] proposed two new techniques for finding intervals based on the mean of the distributions. Abbasov et al. [1] and Tai [46] have built the universal set based on the change of data between consecutive periods of time or their percentage change. Dividing the universal set with appropriate number of groups is an important problem because it will influence the result of the model. Almost all of the existing models give the specific constant in performing. (It is often five or seven.) Others determine it based on the experiments from many data sets. However, the number of groups is only considered suitable if they depend on the similar level of elements in series. When the elements in a series have a lot of difference, the number of found groups will be large and vice versa. In this study, we use whole series as the universal set and determine the number of groups for it by the automatic clustering algorithm. Through this algorithm, in

the series where the similarity of the elements is not the same, the number of clusters found will be different.

For (2), many authors divided the universal set to become the equal intervals. The elements in each cluster were also determined by the  $k$ -mean algorithm [9]. Tai [47] built groups for the elements based on the clustering algorithm. In this study, we proposed the cluster analysis method using genetic algorithm to find the specific elements in each group. For (iii), several important studies have been performed. For instance, Song et al. [42] used the matrix operations and Chen et al. [13] took the fuzzy logic relations. Moreover, many authors [4, 19, 20, 28] used artificial neural networks to determine fuzzy relations.

In addition, the fuzzy relationship based on the triangle and trapezoid fuzzy numbers was also considered in [25]. The relationship based on clustering algorithm is also established by Tai and Nghiep [47]. Many researchers had also used either the centroid method such as [13, 27, 28] or the adaptive expectation method [5, 16, 47] to perform. This article contributes to three stages (1), (2), and (3) for FS model:

For (1), we proposed a new algorithm for finding the appropriate number of groups divided for each series. This value depends on the similar level of objects in series. This method has outstanding advantages in comparison with the existing ones that were presented as linguistic values with levels being constant. (It is usually five or seven in applications.)

For (2), we propose the improved fuzzy genetic algorithm. It can find the specific elements for each group and the probability to belong to the established groups of the element in series.

For (3), based on the principle for normalizing series and the result from (2), a new interpolating method is also proposed.

Incorporating all these improvements, we propose the best model for series time. This model is better than the existing ones through many well-known data sets. We also establish the MATLAB procedure for the proposed model. This procedure can perform effectively for real data. In addition, we also apply the proposed model to forecast the number of COVID-19 victims in Vietnam.

The next section of the paper is structured as follows. Section 2 presents some definitions related to fuzzy time series and proposes a new model. This section also proves the convergence of the proposed model. Section 3 gives the specific steps of the new model and compares it with existing ones over many data sets. An application in Vietnam of the proposed model is presented in Sect. 4. The final section is conclusion.

## 2 The proposed algorithm

### 2.1 The parameters to evaluate the established model

Given a series of historical data  $\{X_i\}$  and predictive value  $\{\hat{X}_i\}$ ,  $i = 1, 2, \dots, N$ , respectively, we have the popular parameters to evaluate the built FTS models as follows:

*Mean squared error:*

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{X}_i - X_i)^2.$$

*Mean absolute error:*

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{X}_i - X_i|$$

*Mean absolute percentage error:*

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left( \frac{|\hat{X}_i - X_i|}{X_i} \cdot 100 \right).$$

*Symmetric mean absolute percentage error:*

$$SMAPE = \sum_{i=1}^N \left( \frac{|\hat{X}_i - X_i|}{(X_i + \hat{X}_i)/2} \cdot 100 \right).$$

*Mean absolute scaled error:*

$$MASE = \frac{\sum_{i=1}^N |\hat{X}_i - X_i|}{\frac{N}{N-1} \sum_{i=2}^N |X_i - X_{i-1}|}.$$

For the built models, the smaller these parameters are, the better the models are.

### 2.2 The proposed algorithm

A cluster with  $m$  elements is given. If these elements converge to the same one  $v$  by any algorithm, then  $v$  is called as the prototype element of the cluster. Let  $T = \{T_1, T_2, \dots, T_N\}$  be the time series, and the  $V^{(t)}$  be set of prototype elements for clusters built at time  $t$ . We propose the forecasting model based on the genetic algorithm and clustering technique as follows:

*Step 1* Initialize  $t = 0$  and  $V^{(0)} = \{v_1^{(0)}, v_2^{(0)}, \dots, v_N^{(0)}\} = \{X_1, X_2, \dots, X_N\}$ . where  $X_c = 10T_c / \max\{T\}$ ,  $1 \leq c \leq N$ .

*Step 2* Update the prototype elements using Formula (1):

$$v_i^{(t+1)} = \frac{\sum_{j=1}^N f(v_i^{(t)}, v_j^{(t)}) v_j^{(t)}}{\sum_{j=1}^N f(v_i^{(t)}, v_j^{(t)})}, 1 \leq i \leq N, \tag{1}$$

where

$$f(v_i^{(t)}, v_j^{(t)}) = \begin{cases} \exp\left(-\frac{d_E(v_i^{(t)}, v_j^{(t)})}{\lambda}\right) & \text{if } d_E(v_i^{(t)}, v_j^{(t)}) \leq \mu\alpha_{ij}(t), \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where  $\alpha_{ij}(t) = \alpha_{ij}(t-1) / [1 + \alpha_{ij}(t-1)f(v_i^{(t-1)}, v_j^{(t-1)})]$  is the balance factor, and  $\alpha_{ij}(0) = 1$ .  $\mu = \sum_{i < j} d_E(v_i^{(t)}, v_j^{(t)}) / \binom{N}{2}$  is the average of Euclidean distance  $d_E(v_i^{(t)}, v_j^{(t)})$ ,  $\lambda = \sigma / r$ ,  $\sigma = \sqrt{\sum_{i < j} [d_E(v_i^{(t)}, v_j^{(t)}) - \mu]^2 / \binom{N}{2}}$  is the standard deviation, and  $r$  is a constant.

**Step 3** Repeat Step 2 until  $\|v^{(t+1)} - v^{(t)}\| = \max_i \{|v_i^{(t+1)} - v_i^{(t)}|\} < \varepsilon$ .

$v^{(t+1)}$  determined by (1) is expansion or narrowing of  $v^{(t)}$ , such that the elements in the same group will be changed to become a prototype element. It means that after an iteration of Step 2, each element in  $X$  will converge to the prototype element of group containing it. Step 3 ends when the difference of all elements between two successive iterations is less than  $\varepsilon$ . This value can affect the number of divided groups as well as the computational cost. The iterations of algorithm will increase if the value of  $\varepsilon$  decreases. In this study, we have taken  $\varepsilon = 10^{-4}$  for all numerical examples.

**Step 4** Encode the clustering solutions. In general genetic algorithm, each variable is represented by a gene, and the chromosome is the set of those genes, which represents a solution to the problem. In the proposed algorithm, the chromosome  $M$  is formed by  $kp$  genes representing for  $k$  clusters.

**Step 5** Initialize  $N$  chromosomes and evaluate their Improved Davies and Bouldin index [18] by (3).

$$IDB = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \left\{ \frac{\frac{1}{|C_i|} \sum_{X_i \in C_i} d_E(X_i, \bar{X}_i) + \frac{1}{|C_j|} \sum_{X_j \in C_j} d_E(X_j, \bar{X}_j)}{d_E(\bar{X}_i, \bar{X}_j)} \right\}, \tag{3}$$

where,

- $\bar{X}_i$  and  $\bar{X}_j$  are the centroid of  $C_i$  and  $C_j$ .
- $d_E(\cdot)$  is the Euclidean distance.
- $|C_i|$  is the number of elements in cluster  $C_i$ .

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

- **Crossover** Perform the crossover operator to chromosomes by the probability 0.85. Let  $L_1$  and  $L_2$  be the two parent chromosomes; then, the child chromosome is created as follows:

$$Child = L_1 + 0.85 \cdot (L_2 - L_1).$$

In a similar way for all chromosomes in population, we have a new population.

- **Mutation** Let  $h$  be the previous value of the mutated gene, the new value  $h'$  of  $h$  is computed as follows:

$$h' = \begin{cases} (1 \pm 2\delta)h & \text{if } h \neq 0, \\ \pm 2\delta & \text{if } h = 0, \end{cases}$$

where  $\delta$  is a random number belonging to the interval  $[0, 1]$ , and the  $+$  or  $-$  sign occurs with equal probability.

- **Selection** The Roulette wheel strategy [33] is used to implement the selection operation. Probability of choosing the  $i$ th element is determined by (4).

$$p_i = \frac{IDB_i}{\sum_{j=1}^N IDB_j}, \tag{4}$$

where  $IDB_i$  is the fitness function of  $i$  and  $N$  is the size of the current population.

**Step 7** Calculate the  $IDB$  index of the chromosomes obtained in Step 6.

**Step 8** Replace the current clustering solution by the new ones having the smaller  $IDB$  index.

Repeat Step 5 to Step 7 until  $iter > maxiter$ , where  $iter$  and  $maxiter$  are the number of current iterations and the required one of the proposed algorithm, respectively.

The parameters of the genetic algorithm used in the proposed model are summed by Table 1.

**Step 9** Let  $\mu_{ic} \in U^{(0)}$  be the result of fuzzy clustering at the first time  $t = 0$ , and  $M = (M_i), 1 \leq i \leq k$  be the optimal

**Table 1** The used parameters of genetic algorithm

Parameter	Value
Population size	100
Encoding variable	Real
Chromosome length	$kp$
Generations	300
Selection operator	Roulette
Crossover probability	0.85
Mutation probability	0.01

central cluster. Establish the first partition matrix with the elements computed by (5).

$$\mu_{ic} = \begin{cases} 1 & \text{if } X_c \in C_i \\ 0 & \text{if otherwise} \end{cases}; 1 \leq c \leq N, 1 \leq i \leq k \quad (5)$$

Find the prototype element of clusters by (6).

$$w_i = \frac{\sum_{c=1}^N (\mu_{ic})^m X_c}{\sum_{c=1}^N (\mu_{ic})^m}, \quad (6)$$

where  $\mu_{ic} \in U^{(t)}$ ,  $1 \leq i \leq k$  is fuzzy probability of  $k$  clusters, and  $w_i$  is the element official centroid of  $k$  clusters.

The value of  $m$  in (6) is the fuzziness degree. When  $m = 1$ , the fuzzy clustering becomes the non-fuzzy clustering. When  $m \rightarrow \infty$ , the partition becomes completely fuzzy with  $\mu_{ic} = 1/k$ . Although [11, 12, 39] had proposed the rules to find the supreme of  $m$ , the best value of  $m$  has not still been determined. Performing a lot of series, we take  $m = 2$  for applications.

*Step 10* Update the new partition matrix  $U^{(t+1)}$ , where each element of  $U^{(t+1)}$  is determined by (7):

$$\mu_{ic} = \frac{d_E(w_i, X_c)^2}{\sum_{j=1}^k d_E(w_j, X_c)^2}, 1 \leq i \leq k, 1 \leq c \leq N, \quad (7)$$

with  $d_E(w_i, X_c)$  being the Euclidean distance of cluster central  $w_i$  and original data  $X_c$ .

*Step 11* Repeat Step 9 and Step 10 until

$$\|U^{(t+1)} - U^{(t)}\| = \max_{ic} \{|\mu_{ic}^{(t+1)} - \mu_{ic}^{(t)}|\} < \varepsilon.$$

*Step 12* Calculate the center ( $M_i$ ) of each cluster and forecast  $Y_c$  according to the following rule:

$$Y_c = \sum_{i=1}^k M^T \mu_{ic}, 1 \leq c \leq N,$$

where  $M^T = (M_i \times \max\{T\}/10)^T$ .

The proposed model includes three phases with 12 steps. Phase 1 has three steps (Step 1 to Step 3) that are used to determine the suitable number of groups to divide the series. In this phase, at the first time, each element is considered to be a cluster. After many iterations, the elements in the same group will converge to the prototype element. It can run many iterations depending on the similar level of elements in series. The result of this phase is the number of groups  $k$  that the series is divided. In many existing models,  $k$  is often chosen as constant ( $k = 5$  or  $k = 7$ ). In the proposed model,  $k$  is determined by the automatic algorithm.

Phase 2 has five steps (Step 4 to Step 8) that it finds the elements for  $k$  groups by the improved genetic algorithm. It builds the operators such as selection, crossover, and mutation with the objective function as *IDB*. First, the algorithm encodes the clustering solutions. Each variable is

represented by a gene, and the chromosome is the set of those genes, which represents a solution to the problem. Initialize  $N$  chromosomes and evaluate their *IDB*; after that, perform the selection, crossover, mutation operators, and compute again *IDB*. These processes are repeated until the *IDB* index is almost unchanged. In our experiment, this phase often converges with the number of iterations less than 50.

Phase 3 includes three steps (Step 9 to Step 12). It builds the fuzzy relationship of each element in series to the established groups from Phase 2 and proposes the rule to interpolate data. In this phase, Step 9 initializes the initial fuzzy relationship by (5). After that, the elements in this matrix will be updated until all the values of two consecutive iterations are almost unchanged. From the result of this matrix, we give the rule to forecast.

To sum up, the proposed model has to perform three phases, and each phase includes many steps and runs many iterations. Therefore, the computation of the proposed algorithm is complex compared to others. The flowchart of the proposed model is shown in Fig. 1.

We have established the complete MATLAB procedure for the proposed model. It can perform quickly and effectively for real data.

### 2.3 The convergence of the proposed model

The convergence of the proposed model is shown by three phases. Phase 2 will stop when the number of iterations is maxiter. (We chose maxiter = 1000.) The convergence of Phase 3 is similar as the fuzzy cluster analysis algorithm for the discrete elements (FCM) that it has been proven in many documents [31]. Therefore, we only consider the convergence of the Phase 1 which is proved by Theorem 1.

**Theorem 1** *If the function  $f(u, v)$  in (2) satisfies:*

- (i)  $0 \leq f(u, v) \leq 1$  and  $f(u, v) = 1$  when  $u = v$ .
- (ii)  $f(u, v)$  depends only on  $\|u - v\|$ ; the distance from  $u$  to  $v$ .
- (iii)  $f(u, v)$  is decreasing for  $\|u - v\|$ .

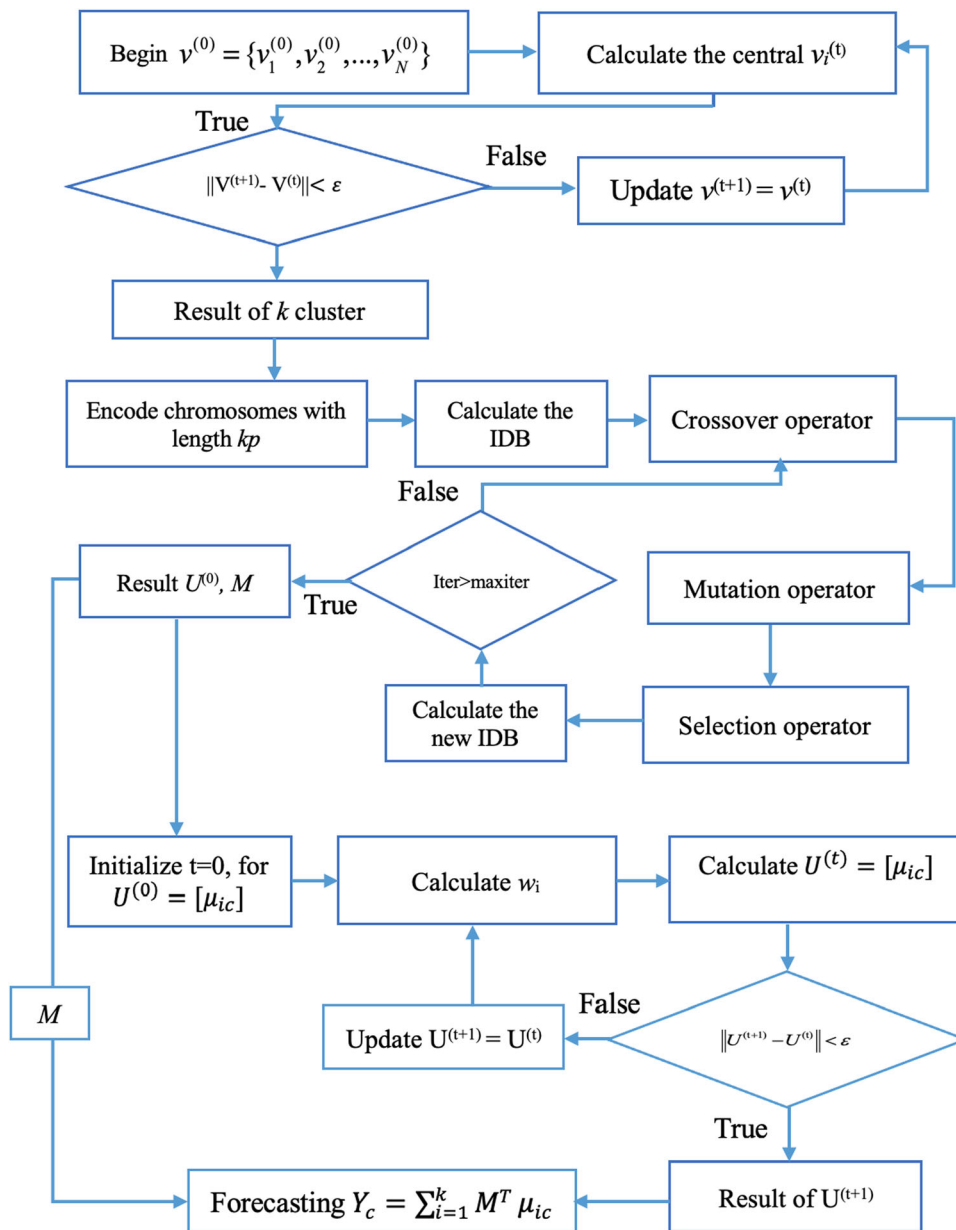
*Then, there exist  $t$  and  $\varepsilon$  such that  $V^{(t)}$  whose elements are determined by (1) satisfies:*

$$\|V^{(t+1)} - V^{(t)}\| < \varepsilon.$$

**Proof** Let  $C_1^{(t)}$  be the convex hull of  $\{v_1^{(t)}, \dots, v_N^{(t)}\}$ . Then,  $v_i^{(t+1)} \in C_1^{(t)}$  is a weighted average of  $v_j^{(t)}, j = 1, \dots, N$ . Therefore,  $C_1^{(t)} \supseteq C\left(\{v_1^{(t+1)}, \dots, v_N^{(t+1)}\}\right) = C_1^{(t+1)}$ . Since

$$C_1 = \lim_{t \rightarrow \infty} C_1^{(t)},$$

**Fig. 1** Flowchart of the proposed model



there exist  $i$  such that  $\lim_{t \rightarrow \infty} u_{1,i}^{(t)} = u_{1,i}$  where  $u_{1,i}^{(t)}$  is a vertex of  $C_1^{(t)}$ . For each  $t$  and  $i$ ,  $u_{1,i}^{(t)} = v_k^{(t)}$  for at least one  $k$ ; there exists  $j$  such that  $v_j^{(t)} = u_{1,i}^{(t)}$  for infinite many  $t$ 's. Therefore, there exists  $t \rightarrow \infty$  such that  $v_j^{(t_n)} = u_{1,i}^{(t_n)}$ , which leads to  $\lim_{n \rightarrow \infty} v_j^{(t_n)} = u_{1,i}$ . We consider two possible cases as follows:

- Case 1: If  $v_j^{(t_n)} = u_{1,i}$  except for any finite  $t$ , then  $\lim_{t \rightarrow \infty} v_j^{(t)} = u_{1,i}$ .
- Case 2: If there exists  $j' \neq j$  and  $s_n \rightarrow \infty$  such that  $\forall n$ ,  $v_{j'}^{(s_n)} = u_j^{(s_n)}$ . Assume that  $u_{1,i}^{(t)} = v_j^{(t)}$  or  $v_{j'}^{(t)}$  for

all  $t > T$ . From equation (2), if  $v_j^{(s)} = v_{j'}^{(s)}$  for some  $s$ ,  $v_j^{(t)} = v_{j'}^{(t)}$  for all  $t > s$ . Therefore, for any  $s > 0$ , there exists  $t > s$  such that  $u_{1,i}^{(t)} = v_j^{(t)}$  and  $u_{1,i}^{(t+1)} = v_{j'}^{(t+1)}$ . We claim that this case, however, can never happen with  $t$  being large enough.

Without loss of generality, assume that  $u_{1,i} = 0$ ,  $v_j^{(t)} \leq 0$ , and  $v_k^{(t)} > 0$  for  $k \neq j$  or  $k \neq j'$ . If  $v_{j'}^{(t+1)}$  later becomes the new vertex, then  $v_{j'}^{(t+1)} < v_j^{(t+1)}$ .

Moreover, since  $v_{j'}^{(t+1)}$  is the new vertex we have

$$v_j^{(t+1)} \leq 0 \Rightarrow \sum_{k=1}^N f(v_j^{(t)}, v_k^{(t)})v_k^{(t)} \leq 0.$$

Since  $v_j^{(t)}$  is the current vertex,  $\|v_j^{(t)} - v_k^{(t)}\| > \|v_j^{(t)} - v_k^{(t)}\|$  for all  $k$ . Then,

$$\sum_{k=1}^N f(v_j^{(t)}, v_k^{(t)})v_k^{(t)} \leq \sum_{k=1}^N f(v_j^{(t)}, v_k^{(t)})v_k^{(t)} < 0,$$

and

$$0 < \sum_{k=1}^N f(v_j^{(t)}, v_k^{(t)}) \leq \sum_{k=1}^N f(v_j^{(t)}, v_k^{(t)}),$$

and we have

$$v_j^{(t+1)} < v_j^{(t+1)},$$

which is a contradiction to (2). Therefore,  $u_{1,i}^{(t)} = v_j^{(t)}$  for some  $j$  and for all  $t$  large enough. Then,  $\lim_{t \rightarrow \infty} v_j^{(t)} = u_{1,i}$ .

We can apply a similar result for  $C_2$  as  $C_1^{(t)}$ ; at least one subject converges to each vertex of  $C_2$ . Then, we can run similar steps again for  $C_3, C_4, \dots$  until all subjects converge. This completes the proof of Theorem 1.  $\square$

### 2.4 The computational complexity of the proposed algorithm

Let  $N$  be the number of elements in series,  $k$  be the number of clusters,  $p$  be the number of dimensions,  $t_{max}$  be the number of iterations of algorithm, and  $P$  be the size of population in the genetic algorithm. Based on the research of Hongchun et al. [26] and Xu et al. [51], the computational complexity of the proposed model is explained as follows:

- \*Phase 1 (Step 1 to Step 3). Because the number of simulation replications is  $t_{max}$ , the computational complexity of this phase is  $O(t_{max}N^2k)$ , where standardizing time series needs  $O(N)$ ; updating the value of prototypes needs  $O(N^2k)$ , and comparing two prototypes in each iteration needs  $O(t_{max}N)$ .
- \*Phase 2 (Step 4 to Step 8). This phase uses the genetic algorithm with some improvements. The computational complexity of genetic algorithm is  $O(t_{max} \cdot NPkp)$  because of the following reasons:

The number of genes in a chromosome is  $kp$ . If there exists variable one in each gene, we need  $O(kp)$  to initialize one gene.  $P$  chromosomes need to be initialized, so the initialization of the whole population needs  $O(Pkp)$ .

In one iteration,  $O(Pkp)$  is required for chromosome crossing. In one chromosome, gene

rearrangement requires  $O(N^2kp)$ .

Each iteration of mutation step is run  $O(Pkp)$  times, and  $O(NPkp)$  is required to select the best chromosomes in each iteration.

The *IDB* index is used as objective functions and required  $O(Nkp)$  to calculate individual fitness.

- \*Phase 3 (remaining steps): It is used to find the fuzzy relationship of elements in clusters, so it has the same construction with fuzzy c-mean algorithm. The computational complexity of this phase is  $O(t_{max}Nkp)$  based on the research of Sreenivasarao and Vidyavathi [45].

In short, the total computational complexity of the proposed algorithm is

$$O(t_{max}N^2Pkp).$$

On comparing this result with other ones, we have Table 2.

Almost all of the popular fuzzy time series models have the computational complexity as the same as the model of Tai (2019) [46]. Table 2 shows that the computational complexity of the proposed model is more complicated than other models.

## 3 Numerical example and comparisons

### 3.1 Numerical example

We use the EnrollmentAU series performed in many studies [13, 46, 47] to illustrate the developed algorithm. The value of the EnrollmentAU series is given by Column  $T_i$  of Table 3.

- Step 1* Initialize  $t = 0$ ; because of  $\max\{T\} = 19337$ ,  $V^{(0)}$  is computed by the third column of Table 3.
- Step 2* Calculating the prototype elements by Formula (1), we obtained  $V^{(1)}$  in Table 3.
- Step 3* Because  $\max_i\{|v_i^{(1)} - v_i^{(0)}|\} = 0.223 > \epsilon$ , the iterations of Phase 1 will continue. After 6 iterations of the above steps, Phase 1 stops. The result of these iterations is given in Table 3 and is shown in Fig. 2.

**Table 2** Comparison of the computational complexity of models

Model	Computational complexity	Assessment
ARIMA	$O(Np)$	Very low
Tai [46]	$O(Nkp)$	Low
Tai and Nghiep [47]	$O(t_{max}N^2kp)$	Medium
Proposed	$O(t_{max}N^2Pkp)$	High

**Table 3** The detailed outcome of the first phase

Year	$T_i$	$V^{(0)}$	$V^{(1)}$	$V^{(2)}$	$V^{(3)}$	$V^{(4)}$	$V^{(5)}$	$V^{(6)}$
1971	13055	6.75	6.76	6.76	6.76	6.76	6.76	6.76
1972	13563	7.02	7.03	7.04	7.05	7.07	7.08	7.09
1973	13867	7.16	7.15	7.14	7.13	7.11	7.10	7.09
1974	14696	7.61	7.61	7.61	7.62	7.62	7.62	7.62
1975	15460	7.99	7.99	7.98	7.97	7.97	7.97	7.97
1976	15311	7.94	7.96	7.97	7.97	7.97	7.97	7.97
1977	15603	8.02	7.99	7.98	7.97	7.97	7.97	7.97
1978	15861	8.20	8.20	8.20	8.20	8.19	8.18	8.18
1979	16807	8.72	8.72	8.72	8.72	8.72	8.71	8.71
1980	16919	8.72	8.72	8.72	8.72	8.72	8.71	8.71
1981	16388	8.48	8.48	8.48	8.49	8.49	8.50	8.51
1982	15433	7.99	7.98	7.98	7.97	7.97	7.97	7.97
1983	15497	7.99	7.99	7.98	7.97	7.97	7.97	7.97
1984	15145	7.87	7.89	7.92	7.94	7.96	7.96	7.97
1985	15163	7.87	7.89	7.92	7.94	7.96	7.96	7.97
1986	15984	8.23	8.22	8.21	8.20	8.19	8.18	8.18
1987	16859	8.72	8.72	8.72	8.72	8.72	8.71	8.71
1988	18150	9.39	9.39	9.39	9.39	9.39	9.39	9.39
1989	18970	9.80	9.80	9.80	9.81	9.81	9.82	9.82
1990	19328	9.99	9.99	9.98	9.98	9.97	9.97	9.96
1991	19337	9.99	9.99	9.98	9.98	9.97	9.97	9.96
1992	18876	9.79	9.80	9.80	9.81	9.81	9.82	9.82

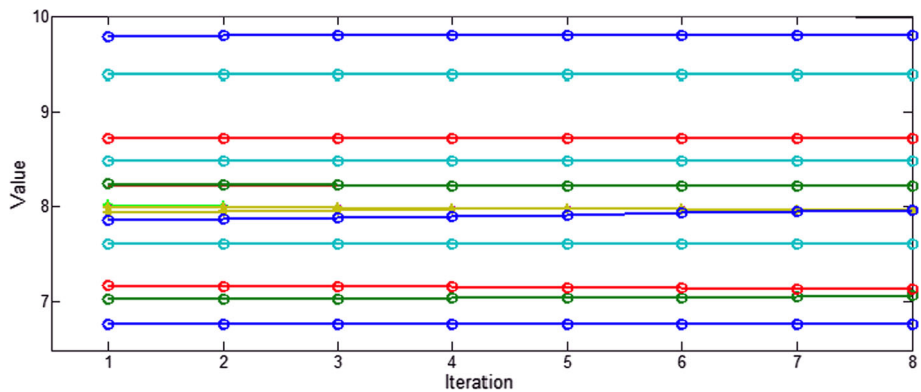
From Table 3 as well as Fig. 2, we see that the elements of series converge to 10 elements. Therefore, we divide the given series to  $k = 10$  groups.

**Step 4** From the result of Phase 1, we encode the first chromosome as follows:

$$M^0 = \{8.768; 6.791; 7.249; 8.136; 7.687; 8.245; 9.942; 9.533; 8.490; 9.737\}.$$

**Step 5** Initialize 100 chromosomes and calculate the objective function (*IDB*) for each chromosome in

**Fig. 2** The convergence of EnrollmentUA data to 10 clusters



population. The values of 100 chromosomes and *IDB* are shown in Table 10 (see Appendix A). Then, selecting the best chromosome with the smallest *IDB*, we have the chromosome 1 (*IDB* = 0.85). This result is used to create the new population for Step 6 and Step 7. Continue to run the steps of Phase 2 until *IDB* unchanged as shown in Fig. 3.

**Step 8** When Step 8 ends, we have the outcomes as follows:

- The value of the best objective function: *IDB* = 0.1534.
- The optimal clusters:

$$C_1 = \{X_5, X_6, X_{12}, X_{13}\}; C_2 = \{X_{11}\};$$

$$C_3 = \{X_9, X_{10}, X_{17}\};$$

$$C_4 = \{X_4\}; C_5 = \{X_{18}\}; C_6 = \{X_{14}, a_{15}\};$$

$$C_7 = \{X_1, X_2, X_3\};$$

$$C_8 = \{X_{19}, X_{20}, X_{21}; X_{22}\}; C_9 = \{X_8\}; C_{10} = \{X_{16}\}.$$

- The optimal centroid of clusters:

$$M^{300} = \{7.995; 8.266; 8.719; 7.171; 7.600; 7.014; 6.751; 8.475; 9.810; 7.832\}.$$

**Step 9** Performing Phase 3, we have the first partition matrix with  $t = 0$  as follows:

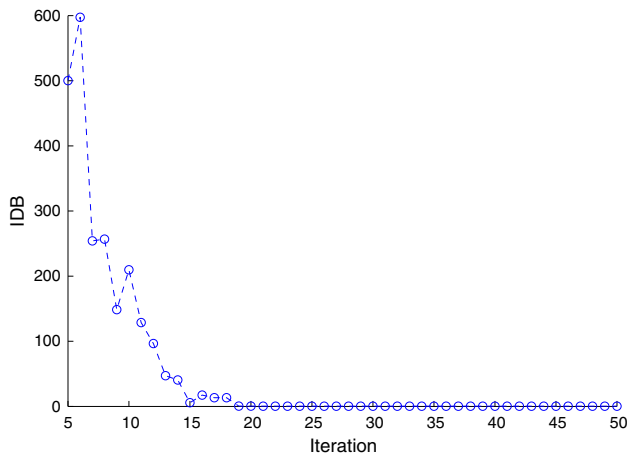
$$\mu_{ic} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Calculating the representative element of clusters, we have:

$$w_i = [7.95; 6.98; 10.00; 9.39; 9.79; 8.20; 8.48; 7.60; 8.27; 8.72]$$

**Step 10** Updating the new partition matrix, we obtain  $U^{(1)}$ :





**Fig. 3** The convergence of the proposed method in Phase 2 after 300 iterations

*Step 11* Repeating Step 9 and Step 10 for  $t = 300$  iterations, Phase 3 stops. At that time, we obtain the matrix  $\mu_{ic}^{(300)}$  as shown in Fig. 4.

*Step 12* Applying the formula

$$Y_c = \sum_{i=1}^{10} M^T \mu_{ic}^{(300)}, 1 \leq c \leq 22,$$

where

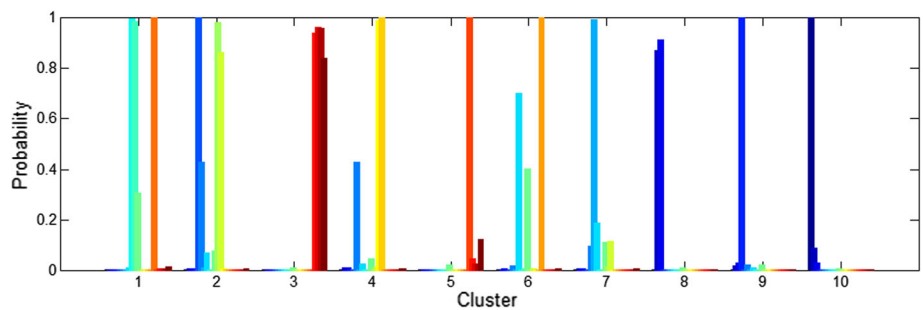
$$M = M^{300} \cdot \max\{T\}/10 = \{15460; 15984; 16859; 13867; 14696; 13563; 13055; 16388; 18970; 15145\},$$

we obtain Table 4 and is illustrated in Fig. 5.

$$\mu_{ic}^{(1)} = \begin{bmatrix} 0.030 & 0.001 & 0.044 & 0.000 & 0.906 & 0.963 & 0.422 & 0.000 & 0.001 & 0.001 & 0.000 & 0.956 & 0.806 & 0.659 & 0.699 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.831 & 0.992 & 0.723 & 0.000 & 0.002 & 0.001 & 0.005 & 0.000 & 0.000 & 0.000 & 0.000 & 0.001 & 0.003 & 0.013 & 0.011 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.004 & 0.000 & 0.003 & 0.000 & 0.000 & 0.000 & 0.002 & 0.000 & 0.000 & 0.001 & 0.000 & 0.000 & 0.001 & 0.002 & 0.002 & 0.000 & 0.000 & 0.000 & 0.016 & 1.000 & 1.000 \\ 0.006 & 0.000 & 0.005 & 0.000 & 0.001 & 0.000 & 0.003 & 0.000 & 0.002 & 0.002 & 0.000 & 0.000 & 0.002 & 0.004 & 0.003 & 0.000 & 0.000 & 1.000 & 0.003 & 0.000 & 0.000 \\ 0.005 & 0.000 & 0.004 & 0.000 & 0.001 & 0.000 & 0.002 & 0.000 & 0.001 & 0.001 & 0.000 & 0.000 & 0.001 & 0.002 & 0.002 & 0.000 & 0.000 & 0.000 & 0.979 & 0.000 & 0.984 \\ 0.020 & 0.001 & 0.025 & 0.000 & 0.042 & 0.012 & 0.335 & 1.000 & 0.003 & 0.003 & 0.000 & 0.019 & 0.093 & 0.067 & 0.063 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.014 & 0.001 & 0.016 & 0.000 & 0.008 & 0.003 & 0.036 & 0.000 & 0.017 & 0.011 & 1.000 & 0.004 & 0.016 & 0.022 & 0.021 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.060 & 0.004 & 0.146 & 1.000 & 0.012 & 0.010 & 0.027 & 0.000 & 0.001 & 0.001 & 0.000 & 0.006 & 0.019 & 0.170 & 0.142 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.019 & 0.001 & 0.022 & 0.000 & 0.025 & 0.008 & 0.154 & 0.000 & 0.004 & 0.004 & 0.000 & 0.011 & 0.052 & 0.049 & 0.046 & 1.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.011 & 0.000 & 0.011 & 0.000 & 0.003 & 0.002 & 0.014 & 0.000 & 0.971 & 0.976 & 0.000 & 0.002 & 0.007 & 0.012 & 0.011 & 0.000 & 1.000 & 0.000 & 0.000 & 0.000 & 0.001 \end{bmatrix}$$

$$\mu_{ic}^{(300)} = \begin{bmatrix} 0.000 & 0.002 & 0.002 & 0.000 & 0.000 & 0.004 & 0.000 & 0.011 & 0.993 & 0.981 & 0.307 & 0.000 & 0.001 & 0.000 & 0.000 & 0.000 & 0.999 & 0.000 & 0.006 & 0.005 & 0.006 & 0.015 \\ 0.000 & 0.006 & 0.008 & 0.000 & 0.997 & 0.427 & 0.008 & 0.067 & 0.001 & 0.002 & 0.074 & 0.979 & 0.860 & 0.006 & 0.000 & 0.000 & 0.000 & 0.000 & 0.002 & 0.002 & 0.002 & 0.005 \\ 0.000 & 0.001 & 0.001 & 0.000 & 0.000 & 0.001 & 0.000 & 0.001 & 0.000 & 0.001 & 0.008 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.936 & 0.959 & 0.955 & 0.836 \\ 0.000 & 0.009 & 0.011 & 0.000 & 0.001 & 0.428 & 0.001 & 0.023 & 0.001 & 0.002 & 0.043 & 0.006 & 0.015 & 0.988 & 0.999 & 0.000 & 0.000 & 0.000 & 0.002 & 0.002 & 0.002 & 0.004 \\ 0.000 & 0.001 & 0.001 & 0.000 & 0.000 & 0.001 & 0.000 & 0.002 & 0.001 & 0.003 & 0.020 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 1.000 & 0.044 & 0.023 & 0.025 & 0.119 \\ 0.000 & 0.004 & 0.004 & 0.000 & 0.000 & 0.019 & 0.001 & 0.697 & 0.002 & 0.006 & 0.401 & 0.001 & 0.007 & 0.001 & 0.000 & 1.000 & 0.000 & 0.000 & 0.003 & 0.003 & 0.003 & 0.007 \\ 0.000 & 0.005 & 0.006 & 0.000 & 0.002 & 0.093 & 0.989 & 0.186 & 0.001 & 0.003 & 0.108 & 0.012 & 0.114 & 0.002 & 0.000 & 0.000 & 0.000 & 0.000 & 0.003 & 0.002 & 0.003 & 0.006 \\ 0.000 & 0.868 & 0.910 & 0.000 & 0.000 & 0.003 & 0.000 & 0.002 & 0.000 & 0.001 & 0.009 & 0.000 & 0.001 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.001 & 0.001 & 0.001 & 0.002 \\ 0.000 & 0.017 & 0.028 & 1.000 & 0.000 & 0.023 & 0.000 & 0.008 & 0.000 & 0.001 & 0.023 & 0.001 & 0.003 & 0.003 & 0.000 & 0.000 & 0.000 & 0.000 & 0.002 & 0.001 & 0.002 & 0.003 \\ 1.000 & 0.087 & 0.029 & 0.000 & 0.000 & 0.002 & 0.000 & 0.001 & 0.000 & 0.000 & 0.006 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.001 & 0.001 & 0.001 & 0.002 \end{bmatrix}$$

**Fig. 4** Bar graph showing the relation between each element with 10 clusters



**Table 4** Forecasting result of the recommended model

Year	Actual	Forecasting	Year	Actual	Forecasting
1971	13055	13073.44	1982	15433	15456.14
1972	13563	13692.32	1983	15497	15469.47
1973	13867	13747.85	1984	15145	15238.63
1974	14696	14735.18	1985	15163	15237.53
1975	15460	15455.77	1986	15984	15937.87
1976	15311	15366.72	1987	16859	16735.23
1977	15603	15587.02	1988	18150	18023.26
1978	15861	15821.93	1989	18970	19152.25
1979	16807	16730.75	1990	19328	19185.84
1980	16919	16723.93	1991	19337	19177.57
1981	16388	16085.19	1992	18876	18969.28

Computing the parameters to evaluate the model, we have

$$MSE = 14087, MAE = 94.90, MAPE = 0.57.$$

Figure 5 shows that the actual values are almost identical with the forecasted ones. It means that the proposed model is very suitable to forecast this series time.

### 3.2 Comparison with the popular models

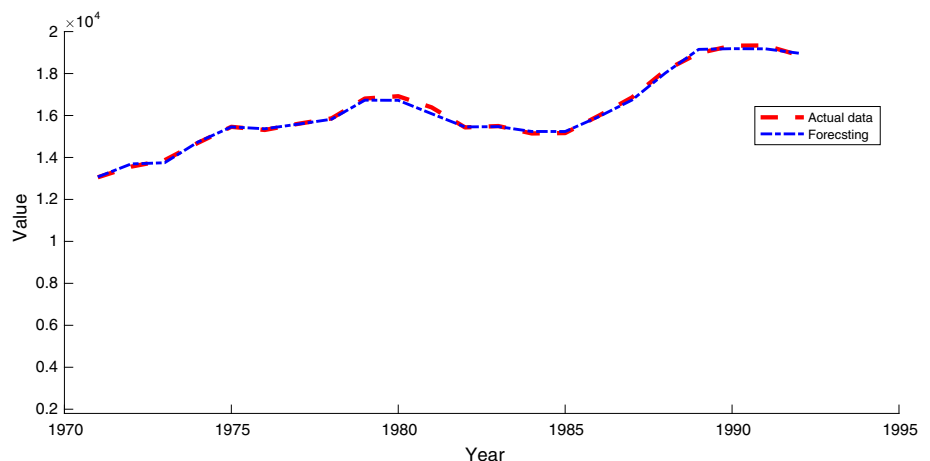
This section compares the new model with the existing ones by the well-known data sets. The considered series are EnrollmentUA [13], Taifex (Taiwan Stock Exchange) [15], Outpatient [23], and Foodgrain [25], and the compared models are Abbasov–Manedova [1] (AM), [34] (L-C), [27] (Hua), [23] (B-R), [41] (Si), [52] (Y-H), [25] (Gh), [13] (Chen), [15] (C-K), [16] (C-H), [32] (Kha), [53] (Yus), [21] (Egr), and Tai [46]. In each datum, we consider two cases:

*Case 1* All of the series are used to build the models and to evaluate them by the parameters MAE, MAPE, and MSE.

*Case 2* Eighty percent of each series is taken as the training set to build the ARIMA (autoregressive integrated and moving average), AM (Abbasov–Manedova), and IFTS [46] models (the others only interpolate) and about 20% of the remaining series is used as the test set. The effectiveness of the model is also evaluated by the MAE, MAPE, and MSE.

- For Case 1, we obtain the results in Table 5: Table 5 shows that the MAE, MAPE, and MSE parameters of the new model are always smaller than the compared existing models for all data sets. The parameters have shown the outstanding advantages in comparing the proposed model to others. For example, the value of MAPE of the proposed model for EnrollmentUA, Taifex, Outpatient, and Foodgrain data sets is 0.57, 0.1, 0.47, and 2.06, respectively, while the others have  $MAPE \in [1.02; 3.08]$  for EnrollmentUA,  $MAPE \in [0.16, 1.42]$  for Taifex,  $MAPE \in [1.09, 24.45]$  for Outpatient, and  $MAPE \in [4.53, 10.13]$  for Foodgrain. We also obtain the similar results for the MSE and MAE parameters.
- For Case 2, their results are given in Table 6 (ARIMAR, AMR, and IFTSR are the ARIMA, IFTS, and Abbasov models with the original data set. ARIMAP, IFTSP, and AMP are the ARIMA, IFTS, and Abbasov models with a training set from the proposed algorithm).

From Table 6, we also obtain the smallest value of the MAE, MAPE, and MSE for the proposed model on comparing with other ones.

**Fig. 5** Line graph of original and forecasting data

**Table 5** Parameters of models for the training sets

Data	Criteria	L-C	Hua	A-M	Si	Gh
EnrollmentUA	MAE	296.15	299.15	479.57	254.16	298.68
	MAPE	2.69	2.45	2.87	1.53	1.82
	MSE	255227	226611	342326	95305	186421
Taifex	MAE	38.27	96.71	89.30	46.01	71.10
	MAPE	0.89	1.39	1.32	0.70	1.03
	MSE	918.16	14391	14136	2968	937
Outpatient	MAE	76.23	96	181	119.03	56.18
	MAPE	11.54	13.75	22.50	2.12	1.98
	MSE	12703	14706	42767	17995.74	16754.35
Foodgrain	MAE	47.76	58.64	89.60	8.69	8.17
	MAPE	6.47	4.53	5.81	5.43	4.98
	MSE	175.43	4772	10672	104.25	123.45
Data	Criteria	C-H	Y-H	Tai	C-K	B-R
EnrollmentUA	MAE	293.45	216.50	168.84	314.34	285.28
	MAPE	1.76	2.15	1.02	2.17	1.65
	MSE	138366.80	47231.03	28525.00	41235	174390.90
Taifex	MAE	11.36	21.32	11.40	25.71	9.27
	MAPE	0.17	1.42	0.17	1.03	0.16
	MSE	230.76	22801	527.81	7679.0	94.65
Outpatient	MAE	107.40	138.38	159.80	167.15	249.17
	MAPE	1.89	2.17	24.45	2.74	3.06
	MSE	16255.32	156.39	37551.87	3890.76	165755.00
Foodgrain	MAE	107.71	67.23	60.35	7.45	7.95
	MAPE	7.01	5.96	4.55	5.21	6.62
	MSE	183.56	2987.15	6460	2345.21	124.07
Data	Criteria	Chen	Yus	Egr	Kha	<b>Proposed</b>
EnrollmentUA	MAE	502.38	182.51	192.15	211.12	94.90
	MAPE	3.08	1.62	1.83	2.12	0.57
	MSE	413980.98	31752	34280	31021	14087
Taifex	MAE	45.24	19.32	21.15	17.18	7.08
	MAPE	0.66	0.78	0.98	0.85	0.10
	MSE	4225.29	824.00	1012	921.15	106.48
Outpatient	MAE	325.96	96.34	86.28	49.98	31.30
	MAPE	5.82	1.34	1.45	1.09	0.47
	MSE	181554.56	3421.24	3017.36	2908.48	162.25
Foodgrain	MAE	16.18	109.15	6.98	5.98	3.05
	MAPE	10.13	7.57	5.09	4.87	2.06
	MSE	440.26	256.57	123.08	98.28	14.60

**Table 6** Parameters of models for the test sets

Data	Model	MAE	MAPE	MSE
EnrollmentUA	ARIMAR	742.27	3.93	901,655.37
	AMR	1,785.28	9.39	3,326,909.30
	IFTSR	414.97	2.20	407512.99
	AMP	750.90	3.98	624155.42
	ARIMAP	723.08	3.81	659,929.11
	IFTSP	412.49	2.19	302182.66
Taifex	ARIMAR	105.37	1.55	12,029.7
	AMR	79.00	1.16	7,117.50
	IFTSR	88.50	1.3	8708750.00
	AMP	36.00	0.53	1,697.31
	ARIMAP	20.16	0.30	786.87
	IFTSP	88.00	1.29	8596.67
Outpatient	ARIMAR	387.11	8.18	247,068.69
	AMR	994.77	20.80	1,439,577.86
	IFTSR	711.10	15.02	831825.25
	AMP	992.27	20.74	1,433,711.74
	ARIMAP	364.14	7.70	227,789.10
	IFTSP	708.38	14.96	817108.16
Foodgrain	ARIMAR	23.29	11.26	656.69
	AMR	15.77	7.91	404.69
	IFTSR	19.47	9.64	535.06
	AMP	13.26	6.72	327.08
	ARIMAP	21.77	10.56	590.10
	IFTSP	10.26	5.20	208.83

To sum up, Tables 5 and 6 show that the proposed model has the best result in both interpolating and forecasting for all considered data sets. This shows the stability and the advantages of the new model. With a lot of considered models, this comparison is very meaningful in evaluating the advantages of the proposed model. In our opinion, the following are the reasons for this result. First, Phase 1 of the proposed model is an automatic algorithm that divides the series into groups with the appropriate numbers based on how similar they are in the series, while the other algorithms give the number of groups according to the experience or by language level. Second, Phase 2 only stops until the IDB index is optimized, while others base on distance criterion and do not consider parameters to evaluate the built algorithm. Finally, the relationship between each element in series and the divided groups is established appropriately. This relationship is built based on the fuzzy clustering algorithm, while others are often established by a specific expression.

### 3.3 Comparison with M3-Competition data

The third competition data called the M3-Competition were expanded from the M1-Competition data and M2-Competition data. It was built by [44]. This is very well known in series time which is often used to compare the efficiency of models together. Data set has 3003 series with many different kinds, including yearly, quarterly, monthly, daily, and others. They also belong to the different areas such as micro, industry, macro, finance, demographics, and other. The specificity about this set is presented in a lot of documents such as [44, 46, 47].

For these data, according to [44], the important models need to compare are ForecastPro, ForecastX, Bj automatic, Autobox1, Autobox2, Autobox3, Hybrid, ETS, and Auto-ARIMA (<https://robjhyndman.com/m3comparisons.R>). In the two recent studies, the authors Tai [46], Tai and Nghiep [47] have shown their outstanding advantages in comparison with all the above models. Therefore, we only compare the proposed result with that in [46, 47]. Let  $E(\text{MAPE})$ ,  $E(\text{MASE})$ , and  $E(\text{SMAPE})$  be the average of MAPE, MASE, and SMAPE, respectively. The result is presented in Table 7.

Table 7 shows that the new model is more advantages than the compared existing models. With the large number of considered series and the different features for the M3-competition data set, this comparison has shown the outstanding advantages of the new model in the existing ones.

## 4 A real application for COVID-19 victims in Vietnam

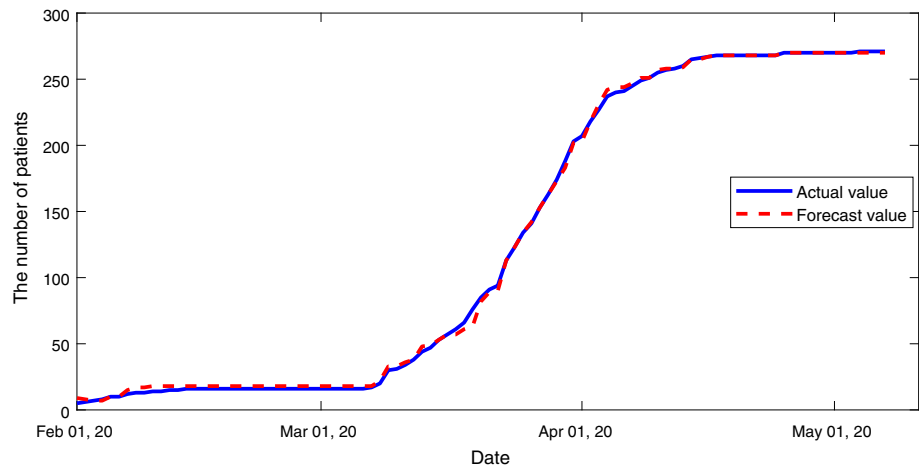
The COVID-19 pandemic is a global problem that most countries in the world are preventing. In the prevention of this pandemic, forecasting the number of victims is one of the important information because it is the base for preventable strategy of the governments. In this section, we use the proposed model to predict the number of COVID-19 victims in Vietnam. It is performed with the following steps:

- The data are divided into two parts: 80% for the training set (97 dates) and 20% remaining for the test set (24

**Table 7** Value of  $E(\text{MAPE})$ ,  $E(\text{MASE})$ , and  $E(\text{SMAPE})$  for models

Methods	$E(\text{MAPE})$	$E(\text{MASE})$	$E(\text{SMAPE})$
Tai [46]	17.31	1.36	10.76
Tai and Nghiep [47]	6.77	1.00	12.76
Proposed model	6.39	0.74	9.20

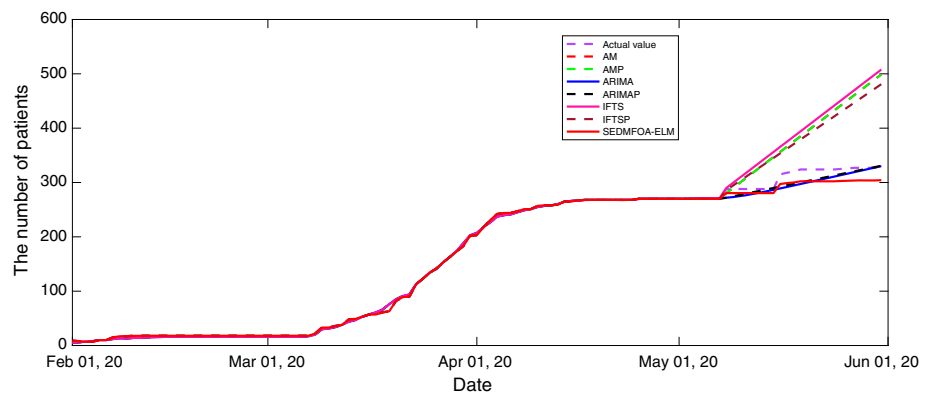
**Fig. 6** Actual and forecasting values for the training set



**Table 8** Comparison of the models of the test set

Parameter	ARIMAR	ARIMAP	AMR	AMP	IFTSR	IFTSP	SEDMFOA [17]
MAE	13.05	11.1	78.29	64.22	86.81	71.53	17.05
MAPE	4.17	3.54	24.44	20.04	27.17	22.37	5.36
MSE	241.26	182.92	8649.12	5957.72	10139.09	7047.29	337.91

**Fig. 7** Forecasting values for test set of models



dates). Interpolating the training set by the proposed model, we have the results shown in Fig. 6. The proposed model has parameters MAE = 1.691; MAPE = 7.169; MSE = 6.288; SMAPE = 0.483; and MASE = 0.475.

Figure 6 shows that the forecasting and actual values are almost identical.

Using the original and interpolated data from the training set to forecast for 24 dates by ARIMA, AM, SEDMFOA [17], and IFTS models, we obtain Table 8 and Fig. 7.

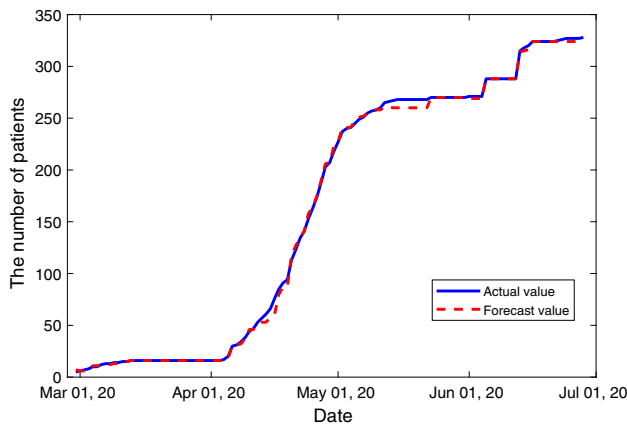
Figure 7 and Table 8 show that the models built from the interpolated data by the proposed model are better than the models built from the original data. Among them, ARIMAP gives the best result with 3.54% of

MAPE, 11.1% of MAE, and 182.92% of MSE. These results have outstanding advantages compared to other models. Therefore, we use this model to forecast for the future.

- Interpolating all data by the proposed model, we have Fig. 8, with MAE = 1.89; MAPE = 2.38; MSE = 11.10; SMAPE = 0.87; and MASE = 0.55.

Using the data from Fig. 8, forecasting for the next several years by ARIMP model, we obtain Table 9.

Table 9 shows that in the next days, the number of COVID-19 victims in Vietnam is barely increased. This is suitable in reality in Vietnam.



**Fig. 8** Interpolating all data by the proposed model

**Table 9** Forecast for the COVID-19 victims for the next 15 days

Date	Number of victims	Date	Number of victims
31 May	324	7 Jun	333
1 Jun	325	8 Jun	334
2 Jun	326	9 Jun	336
3 Jun	327	10 Jun	338
4 Jun	328	11 Jun	340
5 Jun	330	12 Jun	341
6 Jun	331	3 Jun	343
		14 Jun	345

### 5 Conclusion

This research has significant contributions for application of unsupervised learning in building the forecasting model for time series. From the automatic fuzzy genetic algorithm

in clustering, we have some important improvements for the new model. They are the method to find the number of groups to divide the universal set, the algorithm to determine the probability to belong to the divided groups of each element in series, and the principle to interpolate the series from the above result. Implementing for 3007 series with very different numbers and characteristics, the proposed model has shown the stability and has given advantages in comparison with the existing ones via the parameters such as MAPE, MASE, and SMAPE.

A significant contribution of this study is the prediction of COVID-19 victims in Vietnam. Performance results show that the proposed model is good forecasts on this data set. By developing a predictive model that is entirely based on the relationship among the data in the series, we think that the proposed model can get relevant results in predicting COVID-19 victims in other countries. This research can contribute to the early warning of COVID-19 infection risk. This is also our next application direction in the near future.

For this study, we have also faced the problem of computing. Compared with other popular models, the calculation in the proposed model is more complicated. The proposed model has 12 steps that are divided into three phases and are set up in a model. As a result, the time cost of the proposed model is often more than others. In addition, in this study, we are only interested in the optimization of algorithm.

### Appendix A

See Table 10.

**Table 10** The chromosomes are created by the operators

No.	Chromosome										IDB
1	8.28	9.40	8.50	7.79	6.93	8.87	8.58	7.23	6.79	9.82	0.85
2	8.73	7.80	9.34	9.83	8.28	7.15	7.54	8.22	8.60	7.03	1.17
3	9.26	7.81	8.75	8.55	8.66	8.14	9.95	7.87	7.44	7.64	1.32
4	8.04	8.09	6.84	7.96	9.43	9.87	7.98	8.31	7.43	9.79	2.02
5	7.93	9.79	9.37	8.78	7.04	8.11	8.07	9.77	7.19	8.47	2.87
6	6.85	7.80	9.34	7.73	8.28	8.88	8.62	8.22	8.04	7.03	55.32
7	7.96	7.81	8.75	8.55	6.76	8.14	9.95	7.87	7.29	7.15	218.30
8	8.28	7.24	8.87	7.61	9.19	8.87	9.58	9.13	6.79	9.82	1564.09
9	8.47	8.61	9.37	8.77	7.04	8.11	9.36	8.77	9.88	8.39	2148.87
10	9.74	9.05	8.05	8.92	8.07	8.44	7.11	8.48	9.72	7.08	80.55
11	9.26	6.91	8.51	8.55	9.31	8.82	9.91	8.64	9.23	9.69	716.62
12	8.49	9.43	8.76	8.55	9.29	8.24	8.33	7.50	9.88	7.48	259.20
13	8.25	8.35	9.53	7.50	7.15	8.98	7.86	7.27	9.29	7.04	100.65

**Table 10** (continued)

No.	Chromosome										IDB
14	9.26	7.24	9.73	8.55	7.77	7.88	9.95	7.87	7.44	7.64	163.29
15	9.26	9.83	8.75	8.55	8.66	8.88	9.95	9.10	7.44	8.58	447.20
16	9.26	7.81	7.07	7.18	8.85	8.14	9.95	8.13	7.44	7.64	78.56
17	7.37	7.39	9.29	9.03	7.39	8.45	6.83	8.17	9.78	8.42	2438.67
18	9.74	7.17	9.86	7.88	7.76	9.45	8.68	9.10	7.26	8.48	121.07
19	6.86	7.83	8.05	7.26	9.53	7.16	8.22	8.98	8.44	6.98	102.81
20	8.28	7.27	8.50	7.79	6.93	8.17	9.06	7.23	9.28	8.57	272.08
21	8.94	9.40	8.82	8.22	7.01	7.10	7.06	8.92	9.61	8.16	912.82
22	8.30	8.22	9.83	8.01	7.97	9.46	7.53	7.23	8.74	8.25	1.80
23	6.96	9.35	7.62	8.60	7.15	9.41	7.86	8.70	9.45	8.00	164.32
24	9.26	8.03	7.36	7.20	8.33	7.92	9.16	9.30	9.64	9.84	243.81
25	6.75	8.81	7.81	7.79	9.74	7.57	8.58	9.10	9.52	7.85	467.50
26	9.26	9.71	7.36	7.14	9.85	8.79	9.16	8.19	7.29	9.90	209.32
27	9.74	9.05	8.05	8.92	8.07	7.28	7.11	7.79	8.78	7.40	218.66
28	7.96	8.53	9.37	8.43	7.04	8.11	8.12	8.72	7.29	8.47	285.07
29	8.73	7.59	8.90	8.24	9.72	7.15	7.54	6.82	7.03	7.03	1416.39
30	8.73	9.40	9.34	9.83	8.28	8.87	8.58	7.23	6.79	7.03	196.18
31	8.12	8.25	9.94	7.58	8.88	7.07	7.62	7.95	7.62	9.70	967.83
32	6.98	8.84	8.19	8.72	8.50	8.74	6.80	8.26	7.69	9.45	125.14
33	8.34	8.81	7.48	7.86	7.96	7.91	7.20	7.10	9.07	9.27	62.79
34	9.45	6.87	7.01	6.75	6.75	8.27	9.04	6.75	7.67	8.45	125.14
35	7.70	7.76	9.09	7.37	8.57	6.95	9.70	8.69	8.36	8.66	239.80
36	7.94	9.24	7.43	8.46	8.40	7.92	9.67	8.49	9.64	9.02	563.76
37	8.28	8.17	7.35	9.49	6.87	6.97	9.72	7.95	8.61	9.08	43.28
38	9.41	7.80	8.62	9.83	7.32	9.15	7.54	8.22	8.60	7.03	93.76
39	7.37	7.02	7.62	7.38	8.68	9.25	6.83	7.73	7.90	9.22	1568.84
40	7.31	7.62	6.84	7.96	9.33	8.17	7.98	8.31	9.73	9.79	123.77
41	7.69	7.65	7.47	6.97	8.94	8.54	8.36	7.53	9.72	8.70	308.36
42	7.05	6.87	8.40	7.92	9.49	6.83	9.85	7.90	6.82	9.41	906.05
43	6.88	7.84	7.12	6.97	7.66	8.54	9.36	8.27	7.61	8.70	189.66
44	7.19	9.75	8.51	9.84	7.53	8.37	8.60	9.69	7.67	9.15	144.63
45	6.94	7.80	7.49	8.51	8.28	7.15	7.54	7.64	9.78	9.65	359.12
46	8.39	7.25	7.12	6.87	8.66	8.52	9.32	7.25	9.39	9.56	1368.78
47	8.93	9.40	7.76	7.08	7.92	8.87	7.20	6.83	6.79	9.82	304.40
48	9.98	8.97	9.80	7.76	8.82	8.74	6.83	7.94	6.84	7.10	567.32
49	8.04	9.79	6.84	7.96	9.43	8.11	8.07	8.31	7.19	9.79	56.07
50	7.31	7.62	7.13	7.79	9.33	8.17	8.58	7.23	9.73	8.43	242.92
51	7.94	7.30	8.76	9.71	9.98	6.78	8.58	8.23	9.23	7.18	1.06
52	8.22	7.47	8.87	8.41	6.82	6.89	7.86	8.78	7.75	8.59	153.46
53	9.41	7.70	8.62	8.25	7.32	9.15	8.69	8.95	9.48	9.46	681.49
54	8.28	9.40	7.35	7.79	8.45	9.25	8.58	7.23	6.79	9.82	131.87
55	8.73	7.80	9.47	9.83	9.96	7.15	8.15	9.01	8.60	7.03	55.17
56	7.92	8.58	7.38	7.59	6.97	6.76	8.05	9.77	9.00	7.73	112.55
57	6.76	9.58	9.00	9.46	7.62	8.44	9.98	7.61	7.27	9.69	117.63
58	10.00	7.59	8.78	8.23	9.98	8.63	7.70	9.63	8.63	8.32	138.49
59	8.30	8.22	7.12	7.85	7.66	7.97	8.82	7.23	8.36	8.25	245.95
60	9.89	8.12	8.61	6.92	8.37	7.02	8.67	8.66	9.47	9.27	3044.91
61	9.41	9.78	8.62	9.18	8.06	9.15	9.10	8.86	9.69	9.02	741.95

**Table 10** (continued)

No.	Chromosome										IDB
62	9.41	7.70	8.62	8.25	7.32	8.11	8.07	7.69	8.72	9.46	216.79
63	8.91	7.95	7.12	9.34	9.74	9.39	9.36	7.67	8.36	9.79	686.66
64	8.88	9.59	9.49	7.12	7.82	8.83	7.78	7.02	7.49	8.43	342.89
65	7.26	6.91	6.84	9.98	9.43	8.82	9.91	8.31	9.23	9.69	64.56
66	9.29	8.63	8.05	9.38	6.77	8.73	8.83	8.83	9.60	6.98	7231.62
67	9.74	8.58	8.05	7.59	8.07	6.76	8.05	8.48	8.78	7.08	2143.53
68	7.92	7.36	9.09	7.37	8.62	6.90	6.81	9.77	8.99	7.58	512.28
69	8.30	7.86	9.29	9.98	8.51	8.52	6.98	7.08	9.10	6.89	60.90
70	6.87	8.55	9.74	7.85	7.66	7.97	9.36	8.27	8.36	9.79	113.65
71	6.94	9.46	8.64	8.51	8.07	7.07	9.70	7.66	9.78	9.65	330.02
72	7.71	8.84	8.45	7.76	8.51	8.67	6.83	9.62	9.87	7.75	621.27
73	10.00	7.16	7.78	9.21	8.31	9.86	9.16	8.76	8.63	8.85	232.44
74	8.65	9.59	7.89	8.69	7.28	7.50	8.84	9.98	9.00	7.52	506.96
75	8.73	8.97	9.80	9.83	8.82	7.15	7.54	8.22	6.84	7.03	114.13
76	9.29	9.56	9.47	7.18	8.85	9.52	7.48	8.13	8.33	7.32	355.48
77	7.63	9.46	8.64	7.30	8.53	7.07	7.23	9.20	9.18	9.55	740.69
78	9.48	7.83	8.46	7.24	8.64	6.81	7.38	9.28	8.44	9.70	60.69
79	9.25	7.76	9.78	8.01	7.97	8.77	7.33	6.77	9.07	9.97	67.64
80	6.76	7.07	9.55	8.51	8.07	8.33	7.17	7.64	7.27	8.05	109.01
81	8.65	9.40	6.85	9.59	6.95	6.79	6.85	8.97	8.79	8.53	5012.21
82	8.49	9.43	9.09	7.37	8.08	9.79	8.33	9.77	8.36	8.04	304.80
83	9.43	9.84	9.63	7.01	9.26	6.78	8.78	8.20	7.90	9.09	147.56
84	8.98	9.26	9.55	7.34	9.76	9.49	9.76	8.38	7.28	7.85	16581.14
85	8.12	8.25	7.81	8.90	9.81	7.57	7.04	9.10	9.52	7.85	60.97
86	8.28	7.91	9.78	7.79	7.72	9.48	7.33	7.23	6.79	9.82	101.73
87	8.36	7.55	9.56	9.72	9.32	7.34	9.57	7.82	7.78	7.36	2828.56
88	7.48	9.71	7.50	8.65	8.66	7.36	8.34	7.10	8.22	7.13	5657.34
89	8.28	9.40	9.50	7.79	6.93	8.87	8.58	7.23	6.79	9.82	91.69
90	8.83	7.90	9.59	7.64	8.93	8.62	8.55	7.26	7.80	9.05	178.78
91	8.68	9.12	8.84	8.33	7.49	7.36	9.54	8.71	8.74	9.70	156.67
92	6.75	7.59	8.90	8.24	9.72	9.04	8.55	6.82	8.03	7.28	71.17
93	7.94	7.30	8.76	8.71	9.98	7.40	8.58	8.23	7.80	7.18	187.30
94	8.26	7.91	9.51	9.98	9.31	9.82	8.91	8.64	9.23	9.69	228.35
95	9.48	9.58	8.35	8.24	9.64	8.20	7.38	8.28	9.02	8.64	436.54
96	7.92	8.36	9.09	8.37	8.62	7.90	6.81	9.77	7.36	9.04	181.86
97	8.04	8.09	6.84	7.96	9.43	8.87	7.98	8.31	7.43	9.79	1.66
98	9.45	6.87	7.01	9.63	8.53	8.27	9.04	6.84	7.67	8.45	265.73
99	7.05	6.87	9.40	9.34	7.33	6.83	9.85	8.13	6.82	9.66	880.99
100	8.04	8.09	6.84	7.96	9.43	9.87	6.98	9.31	7.43	9.79	3.16

**Acknowledgements** For Tai Vovan, this research is funded by Ministry of Education and Training in Viet Nam under grant number B2022–TCT–03.

## Declarations

**Conflict of interest** The authors declare that they have no conflicts of interest.

## References

1. Abbasov A, Mamedova M (2003) Application of fuzzy time series to population forecasting. *Vienna Univ Technol* 12:545–552
2. Abreu PH, Silva DC, Mendes-Moreira J, Reis LP, Garganta J (2013) Using multivariate adaptive regression splines in the construction of simulated soccer teams behavior models. *Int J Comput Intell Syst* 6(5):893–910



3. Al-Douri Y, Hamodi H, Lundberg J (2018) Time series forecasting using a two-level multi-objective genetic algorithm: a case study of maintenance cost data for tunnel fans. *Algorithms* 11(8):123
4. Aladag CH, Basaran MA, Egrioglu E, Yolcu U, Uslu VR (2009) Forecasting in high order fuzzy times series by using neural networks to define fuzzy relations. *Exp Syst Appl* 36(3):4228–4231
5. Aladag S, Aladag CH, Mentis T, Egrioglu E (2012) A new seasonal fuzzy time series method based on the multiplicative neuron model and sarima. *Hacettepe J Math Stat* 41(3):337–345
6. Ali S, Hamid F, Mahdi F, Amir M, Abouzar E (2020) Generation expansion planning in the presence of wind power plants using a genetic algorithm model. *Electronics*. <https://doi.org/10.3390/electronics9071143>
7. Alpaslan F, Cagcag O, Aladag C, Yolcu U, Egrioglu E (2012) A novel seasonal fuzzy time series method. *Hacettepe J Math Stat* 41(3):375–385
8. Alireza B, Ali J, Mojtaba S, Mohammad H, Kwok W (2019) Developing an ANFIS-based swarm concept model for estimating the relative viscosity of nanofluids. *Eng Appl Comput Fluid Mech* 13(1):26–39
9. Bas E, Uslu VR, Yolcu U, Egrioglu E (2014) A modified genetic algorithm for forecasting fuzzy time series. *Appl Intell* 41(2):453–463
10. Baghban A, Jalali A, Shafiee M, Ahmadi MH, Chau KW (2019) Developing an ANFIS-based swarm concept model for estimating the relative viscosity of nanofluids. *Eng Appl Comput Fluid Mech* 13(1):26–39
11. Bora DJ, Gupta AK (2014) Impact of exponent parameter value for the partition matrix on the performance of fuzzy C means Algorithm. *arXiv preprint arXiv:1406.4007*
12. Cannon RL, Dave JV, Bezdek JC (1986) Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE Trans Pattern Anal Mach Intell* 2:248–255
13. Chen SM (1996) Forecasting enrollments based on fuzzy time series. *Fuzzy Sets Syst* 81(3):311–319
14. Chen SM (2002) Forecasting enrollments based on high-order fuzzy time series. *Cybern Syst* 33(1):1–16
15. Chen SM, Kao PY (2013) Taiffex forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines. *Inf Sci* 247:62–71
16. Chen SM, Hsu CC et al (2004) A new method to forecast enrollments using fuzzy time series. *Int J Appl Sci Eng* 2(3):234–244
17. Chen Y, Pi D (2019) Novel fruit fly algorithm for global optimisation and its application to short-term wind forecasting. *Connect Sci* 31(3):244–266
18. Davies DL, Bouldin DW (1979) A cluster separation measure. <https://doi.org/10.1109/TPAMI.%201979.4766909>
19. Egrioglu E, Aladag CH, Yolcu U, Basaran MA, Uslu VR (2009) A new hybrid approach based on sarima and partial high order bivariate fuzzy time series forecasting model. *Exp Syst Appl* 36(4):7424–7434
20. Egrioglu E, Aladag CH, Yolcu U, Uslu VR, Basaran MA (2009) A new approach based on artificial neural networks for high order multivariate fuzzy time series. *Exp Syst Appl* 36(7):10589–10594
21. Egrioglu E, Bas E, Aladag C, Yolcu U (2016) Probabilistic fuzzy time series method based on artificial neural network. *Am J Intell Syst* 6(2):42–47
22. Friedman JH et al (1991) Multivariate adaptive regression splines. *Ann Stat* 19(1):1–67
23. Garg B, Garg R (2016) Enhanced accuracy of fuzzy time series model using ordered weighted aggregation. *Appl Soft Comput* 48:265–280
24. Ghalandari M, Ziamolki A, Mosavi A, Shams Shirband S, Chau KW, Bornassi S (2019) Aeromechanical optimization of first row compressor test stand blades using a hybrid machine learning model of genetic algorithm, artificial neural networks and design of experiments. *Eng Appl Comput Fluid Mech* 13(1):892–904
25. Ghosh H, Chowdhury SP, (2016) An improved fuzzy time-series method of forecasting based on l-r fuzzy sets and its application. *J Appl Stat* 43(6):1128–1139
26. Hongchun Q, Li Y, Xiaoming T (2020) An automatic clustering method using multi-objective genetic algorithm with gene rearrangement and cluster merging. *Appl Soft Comput J*. <https://doi.org/10.1016/j.asoc.2020.106929>
27. Huarng K (2001) Heuristic models of fuzzy time series for forecasting. *Fuzzy Sets Syst* 123(3):369–386
28. Huarng K, Yu THK (2006) Ratio-based lengths of intervals to improve fuzzy time series forecasting. *IEEE Trans Syst Man Cybern Part B Cybern* 36(2):328–340
29. Jamwal PK, Abdikenov B, Hussain S (2019) Evolutionary optimization using equitable fuzzy sorting genetic algorithm. *IEEE Access* 7:8111–8126
30. Jain S, Bisht DC, Singh P, Mathpal PC (2017) Real coded genetic algorithm for fuzzy time series prediction 1897(1):020–021
31. Kamel MS, Selim SZ (1994) New algorithms for solving the fuzzy clustering problem. *Pattern Recognit* 27(3):421–428
32. Khashei M, Bijari M, Hejazi SR (2011) An extended fuzzy artificial neural networks model for time series forecasting 8(3):45–66
33. Lai CC (2005) A novel clustering approach using hierarchical genetic algorithms. *Intell Autom Soft Comput* 11(3):143–153
34. Lee HS, Chou MT (2004) Fuzzy forecasting based on fuzzy time series. *Int J Comput Math* 81(7):781–789
35. Lewis PA, Stevens JG (1991) Nonlinear modeling of time series using multivariate adaptive regression splines (mars). *J Am Stat Assoc* 86(416):864–877
36. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
37. Own CM, Yu PT (2005) Forecasting fuzzy time series on a heuristic high-order model. *Cybern Syst Int J* 36(7):705–717
38. Prashant KJ, Beibit A, Shahid H (2018) Evolutionary optimization using equitable fuzzy sorting genetic algorithm. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2018.2890274>
39. Pal NR, Bezdek JC (1995) On cluster validity for the fuzzy c-means model. *IEEE Trans Fuzzy Syst* 3(3):370–379
40. Sahragard A, Falaghi H, Farhadi M, Mosavi A, Estebarsari A (2020) Generation expansion planning in the presence of wind power plants using a genetic algorithm model. *Electronics* 9(7):1143
41. Singh SR (2007) A simple method of forecasting based on fuzzy time series. *Appl Math Comput* 186(1):330–339
42. Song Q, Chissom BS (1993) Forecasting enrollments with fuzzy time series-Part I. *Fuzzy Sets Syst* 54(1):1–9
43. Song Q, Chissom BS (1994) Forecasting enrollments with fuzzy time series-part II. *Fuzzy Sets Syst* 62(1):1–8
44. Spyros M, Michle H (2000) The M3-competition: results, conclusions and implications. *Int J Forecast* 16:451–476
45. Sreenivasarao V, Vidyavathi S (2010) Comparative analysis of fuzzy C-mean and modified fuzzy possibilistic C-mean algorithms in data mining. *Ijct* 1(1):104–106
46. Tai VV (2019) An improved fuzzy time series forecasting model using variations of data. *Fuzzy Optim Decis Making* 18(2):151–173
47. Tai VV, Nghiep LN (2019) A new fuzzy time series model based on cluster analysis problem. *Int J Fuzzy Syst* 21(3):852–864
48. Taormina R, Chau KW (2015) ANN-based interval forecasting of streamflow discharges using the LUBE method and MOFIPS. *Eng Appl Artif Intell* 45:429–440

49. Teoh HJ, Cheng CH, Chu HH, Chen JS (2008) Fuzzy time series model based on probabilistic approach and rough set rule induction for empirical research in stock markets. *Data Knowl Eng* 67(1):103–117
50. Wu CL, Chau KW (2013) Prediction of rainfall time series using modular soft computing methods. *Eng Appl Artif Intell* 26(3):997–1007
51. Xu Y, Pi D, Yang S, Chen Y (2021) A novel discrete bat algorithm for heterogeneous redundancy allocation of multi-state systems subject to probabilistic common-cause failure. *Reliab Eng Syst Safety*. <https://doi.org/10.1016/j.ress.2020.107338>
52. Yu THK, Huarng KH (2010) A neural network-based fuzzy time series model to improve forecasting. *Exp Syst Appl* 37(4):3366–3372
53. Yusuf S, Mohammad A, Hamisu A (2017) A novel two-factor high order fuzzy time series with applications to temperature and futures exchange forecasting. *Nigerian J Technol* 36(4):1124–1134

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.