Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Improved seagull optimization algorithm of partition and XGBoost of prediction for fuzzy time series forecasting of COVID-19 daily confirmed cases

Sidong Xian, Kaiyuan Chen, Yue Cheng

A R T I C L E   I N F O
Keywords: Improved seagull optimization algorithm, Fuzzy time series, XGBoost, COVID-19

A B S T R A C T
The establishment of fuzzy relations and the fuzzification of time series are the top priorities of the model for predicting fuzzy time series. A lot of literature studied these two aspects to ameliorate the capability of the forecasting model. In this paper, we proposed a new method (FTSOAX) to forecast fuzzy time series derived from the improved seagull optimization algorithm (ISOA) and XGBoost. For increasing the accurateness of the forecasting model in fuzzy time series, ISOA is applied to partition the domain of discourse to get more suitable intervals. We improved the seagull optimization algorithm (SOA) with the help of the Powell algorithm and a random curve action to make SOA have better convergence ability. Using XGBoost to forecast the change of fuzzy membership in order to overcome the disadvantage that fuzzy relation leads to low accuracy. We obtained daily confirmed COVID-19 cases in 7 countries as a dataset to demonstrate the performance of FTSOAX. The results show that FTSOAX is superior to other fuzzy forecasting models in the application of prediction of COVID-19 daily confirmed cases.

1. Introduction

Time series prediction is crucial in many fields, but typical prediction approaches are ineffective for some time series containing fuzzy information. The concept of the fuzzy set, as an extension of set theory, was proposed by Zadeh [1]. Fuzzy sets and their variants are used to deal with problems with fuzzy information. Zeng [2] proposed an intuitionistic fuzzy social network hybrid MCDM model for an assessment of digital reforms in the manufacturing industry in China. The social network multiple-criteria decision-making approach for evaluating unmanned ground delivery vehicles under the Pythagorean fuzzy environment was proposed by Zeng [3]. The fuzzy time series (FST) was first proposed by Song and Chissom [4] on the basis of fuzzy sets to work out time series problems related to fuzzy information. Establishing fuzzy relations and fuzzing the time series are vital steps of the forecasting model in FTS.

Fuzzing original time series is the start of predicting model. With the development of fuzzy time series, there are two main means for dealing with the fuzzification of time series. Partitioning the discourse of the dataset into some intervals is the first method. The fuzzy membership function can transform the original data into the fuzzy membership of intervals, which is the main procedure of this method. The critical point of this method is the length of intervals, and many papers have studied how to split the universe of discourse into appropriate intervals. Researchers [5-8] explored the influence of lengths of intervals and the method of partitioning intervals to improve the performance of forecasting model. Chen [9] proposed the optimal weighting vectors to find optimal partitions. Bose [10] researched the data partitioning and rule selection technique for FTS with the effect of interval length. Lu [11] used interval information granules to improve forecasting in fuzzy time series. Some literature uses intelligent optimization algorithms to partition intervals. Nizam [12] proposed an improved model based on FTS and PSO for forecasting blood glucose level. An improved genetic algorithm was proposed by Bas [13] for predicting fuzzy time series. Tinh [14] researched the prediction of fuzzy time series with particle swarm optimization. In order to partition intervals, an improved artificial fish swarm optimization algorithm was proposed by Xian [15]. In recent years, some new optimization algorithms have emerged, such as seagull optimization algorithm (SOA). They have been shown to have better performance, but have not been used in FTS forecasting models.

Research paper

Abbreviations: ISOA, Improved seagull optimization algorithm; FST, Fuzzy time series; XGBoost, Extreme Gradient Boosting Tree; COVID-19, Corona Virus Disease 2019.

E-mail addresses: sidx@163.com (S. Xian), 910470585@qq.com (K. Chen), 2978751400@qq.com (Y. Cheng).

https://doi.org/10.1016/j.advengsoft.2022.103212
Received 6 April 2022; Received in revised form 14 June 2022; Accepted 23 July 2022
Available online 1 August 2022
0965-9978/© 2022 Elsevier Ltd. All rights reserved.
The second method of fuzzification is based on fuzzy clustering. The fuzzy membership of varieties can be obtained by fuzzy clustering. On the basis of fuzzy clustering, Aliyev [16] proposed a novel fuzzy time series method for hotel occupancy forecasting. A fuzzy time series forecasting model was proposed by Vovan [17] based on an improved fuzzy function and cluster analysis problems. On the basis of Gustafson-Kessel fuzzy clustering Fan [18] proposed a long-term intuitionistic fuzzy time series predicting model. The length of intervals obtained by the optimization method and fuzzy clustering is often different because the distribution of the sample data on its universe is not uniform, and there has a complex internal structure. The appropriate interval length can better reflect the data structure and improve the prediction accuracy.

The establishment of fuzzy relations is the core step of the fuzzy forecasting model. Some literature further studied the fuzzy relation for increasing the accuracy of the forecasting model. Abhishekh [19] researched the fuzzy relations in the forecasting model. Kokca [20] used an ARMA-type recurrent Pi-Sigma artificial neural network to replace fuzzy relations in high-order fuzzy time series. Cheng [21] forecasted the financial market with the weighted association rule and fuzzy time series model. Some other literature uses regression methods or Markov chains instead of fuzzy relations. Alyousifi [22] proposed a fuzzy time series model with Markov chains. A novel forecasting method in fuzzy time series with stochastic seasonal was proposed by Guney [23]. On the basis of fuzzy c-regression, Dincer [24] proposed a novel means to predict fuzzy time series. Zhang [25] proposed a novel predicting method in fuzzy time series based on time series clustering and multiple linear regression. A method that can better describe the trend of time series is critical to improving the accuracy of the predicting model. Fuzzy relation, hidden Markov chain, and linear regression are all trying to find out the changing time series trend.

There are two problems with the fuzzy series forecasting model. One is the optimization algorithms of partition used in FTS are outdated. Some new optimization algorithms such as SOA have been proved to be more accurate, but they are not used in FTS. The other is the result of fuzzy relations is not accurate enough, and linear regression performs poorly on nonlinear time series. Correspondingly, we propose two methods to solve the above two problems. Firstly, Dhiman [26] proposed SOA and it has better accuracy than traditional optimization algorithms, such as PSO. Inspired by SOA, an improved SOA (ISOA) is proposed by enhancing the convergence ability with the Powell algorithm and a random curve action. In this paper, ISOA is used to partition the domain of discourse to get more suitable intervals. Secondly, since Chen [27] proposed Extreme Gradient Boosting Tree(XGBoost), XGBoost has become one of the most popular machine learning methods and obtains impressive achievements in numerous algorithm competitions. XGBoost is a nonlinear model and has outstanding performance in a variety of tasks, which is why we use XGBoost instead of fuzzy relations to forecast the change of fuzzy membership. With the advantages of ISOA and XGBoost, this paper has the following innovations.

1. The ISOA is put forward for better accuracy and convergence ability and applied to get appropriate intervals.
2. It is the first application of XGBoost on fuzzy time series to forecast the change of fuzzy membership in the literature.
3. A new fuzzy time series forecasting model(FTSOAX) is proposed based on ISOA and XGBoost. Meanwhile, FTSOAX is applied to forecast the daily confirmed of COVID-19.

The rest of the paper is as follows. In Section 2, we introduce the preliminary knowledge and basic concepts of the fuzzy set and the fuzzy time series. Then, we proposed an improved seagull optimization algorithm in Section 3. In Section 4, we introduce the symmetric triangular fuzzy membership function and describe the steps of FTSOAX. Finally, we give an application to convince that FTSOAX has better performance than other models in the application of prediction of COVID-19 daily confirmed cases and summarize the contribution of this paper as a conclusion in Section 5 and Section 6, respectively.

2. Preliminaries

2.1. Fuzzy time series

To deal with some problems with fuzzy information, Zadeh first defined the concept of fuzzy set. Song and Chissom proposed fuzzy time series based on fuzzy sets to deal with time series with fuzzy information. In the coming, we will review the concept of fuzzy sets and fuzzy time series.

Definition 1. Let \( U = \{u_1, u_2, \ldots, u_n\} \) and \( U \) is universe of discourse. Fuzzy set \( A \) in \( U \) is the following.

\[
A = f_s(u_i)/u_i + f_s(u_i)/u_i + \cdots + f_s(u_i)/u_i
\]

\( f_s : U \rightarrow [0, 1] \) is the membership function of the fuzzy set \( A \), and \( f_s(u_i) \) denotes the grade of the membership of \( u_i \) in the fuzzy set \( A \).

Definition 2. Let \( Y(t) = \{1, 2, 3, \ldots\} \), a subset of real numbers, be the universe of discourse on which fuzzy sets \( f(t) = \{1, 2, 3, \ldots\} \) is defined, and \( F(t) = \{f_1(t), f_2(t), \ldots\} \). Then \( F(t) \) is called a fuzzy time series defined on \( Y(t) \).

Definition 3. Both \( F(t) \) and \( F(t-1) \) are fuzzy sets. Let \( R(t) \) be the fuzzy relations defined from \( F(t-1) \) to \( F(t) \), and \( R(t) = F(t-1) R(t, t-1) \), where \( * \) stands for composite operation, then \( F(t) \) is said to be derived from \( F(t-1) \) by fuzzy relation \( R(t) \).

Definition 4. Let \( F(t-1) = A_i \), then a fuzzy logic relations (FLR) \( A_i \rightarrow A_j \) can be used to represent the relations between two continuous observations \( F(t-1) \) and \( F(t) \), and \( A_i \) and \( A_j \) indicate the current state and next state of the fuzzy relations, respectively. Assume fuzzy logical relations (FLRs) such that \( A_i \rightarrow A_j \), \( A_j \rightarrow A_k \), \ldots, \( A_l \rightarrow A_p \), so these fuzzy logical relations (FLRG) can form a whole as \( A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow \ldots \)

Definition 5. There is an obvious relations between \( F(t) \) and \( F(t-n) \). \( F(t-n+1), \ldots, F(t-1), \) then \( G(F(t-n), F(t-n+1), \ldots, F(t-1)) = F(t) \) represents \( n \)-th order fuzzy time series forecasting model, where \( G \) is a multivariable function.

2.2. Classical fuzzy time series forecasting model

According to the above definition, the building of the forecasting model in fuzzy time series can be split into four steps. Taking the enrollment of the University of Alabama as the experimental dataset, we will review the classical forecasting model in fuzzy time series.

Step 1. Determine the fuzzy set and fuzzy membership function according to the training set and partition the universe of discourse.

First, we need to find the maximum and minimum values of the training data, and determine the scope of the universe of discourse \( U \). So the universe of discourse should be \( U = [13055, 19337] \), because we can know that the minimum and maximum values are 13055 and 19337 in the data set, respectively. In addition, the range of the \( U \) is often rounded down and up for the convenience of discussion and calculation. In this example, we define \( U = [13090, 20000] \).

The next stage is to partition the defined universe of discourse \( U \). Generally speaking, the range of the divided universe of discourse \( u_i \) should not be too narrow, because of the fuzziness of this problem. In this case, we take 1000 as the interval length, and partition \( U \) into 7 subsets, \( U = (u_1, u_2, \ldots, u_7) \), such as \( u_1 = [19000, 20000], u_2 = [13000, 14000] \)

Next, we need to define the fuzzy membership function. There are many functions for us to choose from, such as trapezoidal, Gaussian, and triangular fuzzy membership functions, etc.
3. An improved seagull optimization algorithm

The swarm intelligence algorithm is a machine learning method that optimizes problems by simulating group behavior. For example, the ant colony optimization algorithm is a representative swarm intelligence algorithm. The seagull optimization algorithm (SOA) has the property of colony optimization algorithm is a representative swarm intelligence algorithm. In this section, an improved seagull optimization algorithm will be proposed by improving the SOA.

3.1. Seagull optimization algorithm

Seagulls are seabirds all over the world, living in groups and using their wisdom to find and attack prey. Migration and aggressive behavior are significant features of seagulls. Seagulls migrate from one place to another according to the change of seasons, looking for the richest source of food to get sufficient vitality. During the migration, each seagull is located in a different position to avoid collision. In a group, seagulls change their position by moving in the orientation of the best location. Dhaman proposed a seagull optimization algorithm based on seagull behavior. The core process of the algorithm is the following.

Step 1. Migration (global search)

SOA simulates how a flock of seagulls moves from one place to another and avoids collisions in the migration process. In addition, SOA calculates the new position with an additional variable A to prevent clashes with neighbors (other seagulls).

\[ C_i(t) = P_s(t) + A \]

(1)

where \( t \) indicates the present iteration, and \( P_s(t) \) is the seagull’s present location. \( C_i(t) \) indicates a new location without collision with other seagulls and the movement behavior of seagulls in feasible space is expressed as A.

\[ A = f_s - (r \ast (f_s / \text{MaxIteration})) \]

(2)

\( \text{MaxIteration} \) indicates the maximum size of iteration. The value of variable A is controlled by \( f_s \), and \( f_s \) linearly reduces from 2 to 0.

\[ B = 2 \ast A^2 \ast \text{random}(0, 1) \]

(3)

\[ M_i(t) = (P_b(t) - P_s(t)) \ast B \]

1: Input parameters: \( A, B, pop, N, \) and \( \text{MAXiteration} \)
2: \( v \leftarrow 1, u \leftarrow 1, f_c \leftarrow 2 \)
3: while \( t < \text{MAXiteration} \) do
4: \( \text{for} \quad p \leftarrow 1 \text{to} \quad \text{pop} \) do
5: \( \text{fitness}(t) \leftarrow \text{ComputeFitness}(P_s) \)
6: \( r_s \leftarrow \text{random}(0, 1) \)
7: \( \theta \leftarrow \text{random}(0, 2\pi) \)
8: \( u \leftarrow u + e^\theta \)
9: \( A \leftarrow f_c - (t \ast (f_c / \text{MaxIteration})) \)
10: \( B \leftarrow 2 \ast A^2 \ast r_u \)
11: \( M_i(t) \leftarrow B \ast (P_b(t) - P_s(t)) \)
12: \( D_i(t) \leftarrow |C_i(t) + M_i(t)| \)
13: \( \omega \leftarrow \text{random}(0, 1) \)
14: \( W_s(t) \leftarrow A + P_s(t) \ast \cos(2\omega u) \)
15: \( D_i(t) \leftarrow D_i(t) + W_s(t) \)
16: \( P_b(t) \leftarrow P_b(t - 1) + D_i(t) \ast x \ast y \ast z \)
17: \( \text{if} \quad t \% N \equiv 0 \quad \text{then} \)
18: \( P_s(t) \leftarrow \text{Powell}(P_s(t)) \)
19: \( \text{sort(fitness(t)), sort(P_s(t))} \)
20: \( f_b(t) \leftarrow \text{fitness}(t)_{p} \)
21: \( \text{if} \quad f_b(t) \leq f_b(t - 1) \quad \text{then} \)
22: \( P_b(t) \leftarrow P_b(t - 1) \)
23: \( t \leftarrow t + 1 \)
24: return \( P_b(s) \) and \( f_b(t) \)

Algorithm 1. An improved Seagull optimization algorithm.

\( P_b(t) \) represents the best position in all seagulls, and \( M_i(t) \) is the orientation of the \( P_b(t) \). At the same time, in order to balance local search and global search, the parameter B is added to control them.

\[ D_i(t) = |M_i(t) + C_i(t)| \]

(4)

The seagulls reach the new position by moving in the direction of the best position after avoiding overlap. \( D_i(t) \) indicates the new location.

Step 2. Attack (local search)

During the migration, seagulls can unceasingly change the speed of attack and angle. And they keep their height with their wings and weight. When attacking, seagulls move spirally in the sky. The following is the movement action in the \( z, y, \) and \( x \) planes.

\[ r = u + e^\theta \]

\( y = r \ast \sin(\theta) \)

\( z = r \ast \theta \)

\( x = r \ast \cos(\theta) \)

\( \theta \) is a random angle number in the range of \([0, 2\pi]\) and \( r \) is the radius of each helix.

\[ P_s(t) \leftarrow P_b(t) + D_i(t) \ast z \ast y \ast x \]

(6)

\( P_s(t) \) represents the seagull’s attack position.

3.2. An improved seagull optimization algorithm

In reality, when migratory birds migrate, they often do not walk straight direction, mainly because migration routes require the appropriate feeding site. In addition, with hot air rising over the land, it is possible to save energy. Therefore, the birds generally along a curve of movement over the ground, not straight flying over the ocean. This is also true for seagulls. To enhance the global search capabilities of SOA, we give seagulls a random curve action in migratory behavior as follows.

\[ W_s(t) = A + P_s(t) \ast \cos(2\omega u) \]

(7)

\( W_s(t) \) represents the random curve action and \( \omega \) is a random number.
of (0, 1). The random curve action enhances the variety of the seagull population in the global search, and its value will decrease as the size of the iteration rise.

To strengthen the local search capability of SOA, we add the Powell algorithm [28] to the attacking behavior of seagulls. Powell algorithm is simple to calculate and has solid local searchability. However, the Powell algorithm is exceptionally dependent on the initial point configuration. The selection of the initial value directly affects whether the algorithm can converge to the global minimum and even causes the algorithm to fail. Therefore, the position information optimized by the SOA algorithm is used as the initial value of the Powell algorithm to avoid Powell search failure.

\[ P_s(t) = \text{Powell}(P_r(t)) \]  
\[(8)\]

It is obvious that the Powell algorithm will increase the time complexity of SOA. So Powell algorithm is used for local search after every \( N \) iteration. We proposed an improved seagull optimization algorithm (ISOA) based on the above two aspects. The ISOA is shown in Algorithm 1.

### 3.3. Experiments of improved SOA

The experiment is necessary to convince the advantage of the ISOA. Firstly, we compare the original SOA with the improved SOA on two standard test functions \( F_1(x) \) and \( F_2(x) \) in Table 1.

We ran SOA and ISOA 100 times on the test function to better demonstrate their performance because their results have a certain degree of randomness. The amount of iterations there is 100, and the size of the population is 20 for each algorithm. The result of the comparison is shown in Fig. 1. "(a)" and "(b)" are the results of \( F_1(x) \) and \( F_2(x) \), respectively. "Pre-improved SOA" is that only uses the random curve action \( W_s(t) \), and "Improved SOA" is based on random curve action and the Powell algorithm. Comparing ISOA with the original SOA, it is explicitly revealed that it has better accuracy and convergence ability. Furthermore, their main results are shown in Table 2. In Table 2, we compare the best, mean, and standard deviation of different SOA's results. Through comparison, we can intuitively realize that the improved SOA has a better performance than the original SOA in terms of mean, best, and standard deviation of results.

To show the performance of the improved SOA, we select four optimization algorithms as a comparison of four test functions. Yolcu [29] proposed a hybrid fuzzy time series model with single particle swarm optimization (PSO). Sarem [30] proposed Grasshopper Optimization Algorithm (GOA) for solving optimization problems. We compare the performance of SOA and ISOA with GOA in Table 2. In Table 2, we compare the best, mean, and standard deviation of different SOA's results. Through comparison, we can intuitively realize that the improved SOA has a better performance than the original SOA in terms of mean, best, and standard deviation of results.

**Table 1**

| Function | Initial range | Dimension | \( F_{\text{min}} \) |
|----------|---------------|-----------|---------------------|
| \( F_1(x) = \sum_{i=1}^{n} x_i^2 \) | \([-100, 100]^n\) | 30 | 0 |
| \( F_2(x) = \prod_{i=1}^{n} |x_i| + \sum_{i=1}^{n} |x_i| \) | \([-10, 10]^n\) | 30 | 0 |
| \( F_3(x) = \sum_{i=1}^{n} (0.5 + x_i)^2 \) | \([-100, 100]^n\) | 30 | 0 |
| \( F_4(x) = \sum_{i=1}^{n} (10 - 10 \cos(2 \pi x_i)) + x_i^2 \) | \([-5.12, 5.12]^n\) | 30 | 0 |
| \( F_5(x) = -\exp\left(\frac{\sum_{i=1}^{n} \cos(2 \pi x_i)}{\sum_{i=1}^{n} x_i^2}\right) + 20 \exp\left(-\frac{\sum_{i=1}^{n} x_i^2}{4}\right) + \frac{\sum_{i=1}^{n} x_i^2}{40} + \frac{\sum_{i=1}^{n} \cos(2 \pi x_i)}{4} \) | \([-32.2, 32.2]^n\) | 30 | 0 |

**Table 2**

| Algorithm | Ind | \( F_1(x) \) | \( F_2(x) \) |
|-----------|-----|-------------|-------------|
| Original SOA | BEST | 4.99 | 5.80 |
| | MEAN | 13.49 | 10.09 |
| | STD | 3.19 | 1.80 |
| Pre-improved SOA | BEST | 3.02 | 7.29 |
| | MEAN | 7.47 | 4.24 |
| | STD | 2.35 | 1.28 |
| Improved SOA | BEST | 0.00 | 0.36E-3 |
| | MEAN | 9.12E-33 | 0.18E-2 |
| | STD | 1.51E-32 | 0.17E-2 |

![Fig. 1. Comparison between SOA and ISOA.](image-url)
Optimisation Algorithm (GOA). Zhang [31] proposed a fuzzy time series model based on the Genetic Algorithm (GA), and a hybrid forecasting system based on the Differential Evolution (DE) is proposed by Jiang [32]. The above four optimization algorithms are compared with the original SOA and improved SOA. Functions in Table 1 are the standard test function as the evaluation criteria of these optimization algorithms. The number of iterations, population, and dimension are 300, 10, and 20, respectively. The number of dimensions of each test function is in Table 1. Fig. 1 depicts the results of the comparison of optimization algorithms.

(a), (b), (c), (d) represent the results of $F_3(x)$, $F_4(x)$, $F_5(x)$, $F_6(x)$, respectively. At the same time, we run these optimization algorithms 100 times on four test functions. Comparing the results among them, we take three indicators BEST, MEAN, and STD as shown in Table 3.

Through the comparison in Fig. 2, we can find that SOA has the best performance than other optimization algorithms. Especially in "(c)" of Fig. 2, we can notice that both the original SOA and the improved SOA fall into a locally optimal point at the 50th iteration. However, the improved SOA successfully jumped out of the local optimal point in the following iteration, while the original SOA converged at the local optimal point all the time. It shows that the improved SOA has better searchability than the original SOA. Furthermore, the results of improved SOA are better than other algorithms in terms of MEAN, BEST, STD in Table 3. Through the comparison above, we think that the improved SOA has better convergence accuracy.

### 3.4. Computational complexity of ISOA

The complexity of an algorithm is a critical criterion for evaluating its performance. All of the optimization algorithms mentioned above require $O(n_d \times n_p)$ time to initialize, where $n_d$ is the dimension of the objective function, and $n_p$ represents the population size. The time complexity of SOA, PSO, and GOA is $O(N \times \text{Max iteration} \times n_d \times n_p \times o_f)$. It takes $O(N)$ time to simulate the entire procedure, and $\text{Max iteration}$ is the maximum number of iterations. The time complexity of computing the objective function is denoted by $o_f$. The GA and DE algorithms have a time complexity of $O(N \times \text{Max iteration} \times n_d \times n_p \times o_f \times c_s \times m_t)$, where $c_s$ and $m_t$ are the crossover and mutation operators, respectively. The time complexity of ISOA is $O((N + T) \times \text{Max iteration} \times n_d \times n_p \times o_f)$, where $o_p$ represents the time complexity of running Powell’s algorithm and $T$ is a hyperparameter ranging from 0 to 1.

The space complexity of the algorithm is the highest amount of space used at any given point in time. All algorithms mentioned in this work have a space complexity of $O(n_d \times n_p)$.

![Fig. 2. Comparison between ISOA and other optimization algorithms.](image-url)
3.5. Discussion of ISOA in terms of convergence accuracy

According to the results of the experiments, ISOA has a noticeable advantage in terms of convergence accuracy. In this section, the reason why ISOA outperforms other optimization methods in terms of convergence accuracy is explored.

Population diversity is thought to be a significant component influencing the convergence accuracy of swarm intelligence optimization algorithms. Experiments [33–35] demonstrate that excellent population diversity can considerably increase the convergence accuracy of PSO, GA, and DE. The variance of positions is an important metric for measuring population diversity. The high variance of the population position can effectively avoid falling into a local optimum and improve the convergence accuracy. According to formula (7), a random curve action is added to SOA in the global search period to improve its population diversity.

\[
P_{\text{ISOA}}(t) = P_{\text{SOA}}(t) + W_s(t) \\
\text{Var}(P_{\text{ISOA}}(t)) = \text{Var}(P_{\text{SOA}}(t)) + \text{Var}(W_s(t))
\]

(9)

\[P_{\text{ISOA}}(t)\] and \[P_{\text{SOA}}(t)\] represent the seagull’s positions of ISOA, SOA respectively. It is obvious that the ISOA seagull’s positions are produced by adding \[W_s(t)\] to the SOA seagull positions. \[\text{Var}\] denotes the data’s variance. It is simple to prove that the variance of the ISOA seagull’s positions is increased by adding \[\text{Var}(W_s(t))\] to the variance of the SOA seagull’s positions.

\[
\sum_{i=1}^{n} (x_i - \bar{x})^2
\]

(10)

To graphically demonstrate ISOA’s population diversity, formula (10) is solved with ISOA and SOA, and compares the population position of their iterative process. For brevity, the dimension \(i\) in formula (10) is 2, and it is solved in the range \([-100,100]\]. The population size for both ISOA and SOA is 10.

In Fig. 3, ‘(a)’, ‘(b)’, ‘(c)’ represent the population positions of ISOA and SOA with a maximum number of iterations of 100, 500 and 1000, respectively. The blue dot indicates SOA and the yellow dot indicates ISOA. The points in Fig. 3 represent the positions explored by ISOA’s SOA during the iteration. There is no doubt that the number of regions reached by the ISOA and SOA increases with the increase of the maximum number of iterations. Furthermore, it is noticeable that ISOA has reached a wider range of areas than SOA. In Fig. 4, ‘(a)’, ‘(b)’, ‘(c)’ represent the heat map of the SOA’s population positions at the

---

Fig. 3. Positions of ISOA and SOA over multiple iterations.

Fig. 4. Heatmap of positions of ISOA and SOA over multiple iterations.
maximum iterations of 100, 500, and 1000, respectively. It’s worthwhile mentioning that the conjugate gradient or quasi-Newton methods aren’t used to speed up local searches because they require gradient information. The Powell algorithm is simple and does not need to calculate the gradient. The convergence rate of PSO is \[ O\left(\frac{1}{\sqrt{n}}\right) \], according to Quan [36]. The evolutionary algorithms (GA, DE) are considered to converge at the rate of \[ O\left(\frac{1}{\sqrt{n \log n}}\right) \] [37]. The convergence rates of both PSO and evolutionary algorithms are sublinear. The exact convergence rate of the Powell algorithm is unclear, but it is thought to have a linear convergence rate [38], hence its convergence rate is better than that of the PSO and evolutionary algorithms. Because the Powell algorithm is extremely sensitive to the initial point, ISOA is employed to search for the good initial region globally.

In general, the benefits of ISOA come from two factors. The first is that the random curve behavior increases population diversity, which assists in global search. The second is that the Powell algorithm improves the local convergence rate, which helps in local search.

4. A novel fuzzy time series forecasting model

This section will propose a new fuzzy time series forecasting model (FTSOAX) based on an improved seagull optimization algorithm and XGBoost. We employ the improved SOA to partition the universe of discourse \( U \) into several intervals \( u_i \). The original data can then be turned into fuzzy data using the symmetric triangular fuzzy membership function. Each interval’s fuzzy membership is represented by the fuzzy data. To handle issues where fuzzy relations are insufficiently accurate, we employ XGBoost rather than fuzzy relations to anticipate the change in fuzzy membership of each interval. Finally, using the inverse operation of the symmetric triangular fuzzy membership function, the anticipated fuzzy data will be turned into real data. The procedure of FTSOAX is expressed as follows to illustrate the details of FTSOAX.

Step1. Determine the parameters of the model

\( n: n \) is the number of intervals, and its range is usually in (5, 20).

\( p: p \) represents the \( p \)-th order fuzzy time series forecasting model, and its range is usually in (1, 5).

\( \text{pop}: \text{pop} \) represents the population number of improved SOA, and its range is usually in (5, 10).

\( \text{dim}: \text{dim} \) is the dimension of improved SOA, and its range is usually in (1, 5).

Maxiter: The Maximum number of iterations of improved SOA, and its range is usually in (50, 1000).

\( N: N \) represents the Powell algorithm is executed every \( N \) iterations in improved SOA, and its range is usually in (1, 100).

\( \gamma: \gamma \) is a constant of the regularization term of XGBoost, and its range is usually in (0, 0.1).

\( \lambda: \lambda \) is a constant of the regularization term of XGBoost, and its range is usually in (0, 0.1).

Step2. Partition the universe of discourse into intervals with the ISOA

Let \( U \) be the universe of discourse, and \( U \) should be appropriately partitioned into \( n \) intervals \( \{u_1, u_2, ..., u_n\} \) with the ISOA (Algorithm 1). The output of ISOA is \( n − 1 \) split points. The centers \( C \) of each interval are \( \{c_1, c_2, ..., c_n\} \). We apply RMSE (Root Mean Squared Error) as the fitness function of ISOA as follows.

\[
\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}
\]

\( \hat{y} \) represents the forecasted data as a result of XGBoost and \( y \) is the original data.

Step3. Fuzzify the training time series

There is a problem with the traditional fuzzy time series forecasting model that the fuzzification and defuzzification are not inverse operations. This problem reduces the accuracy of the forecasting model. We use the symmetric triangular fuzzy membership function to deal with this problem. This fuzzy membership function can realize the accurate conversion between time series and fuzzy time series. Let \( U = \{u_1, u_2, ..., u_n\} \) as the all intervals of time series, and \( C = \{c_1, c_2, ..., c_n\} \) as the center of each interval. \( p_1 \) and \( p_2 \) are the start point of \( u_i \) and end point of \( u_i \), respectively.

Assuming the number of intervals is 5, this fuzzy membership function can be illustrated in Fig. 5. This fuzzy membership function has the following three advantages.

1. When \( x \) is in the center \( c_i \) of a certain interval \( u_i \), the fuzzy membership \( uA_i \) equals 1.

2. When \( x \) is in the boundary of two intervals \( uA_i \) and \( uA_{i+1} \), both the fuzzy membership \( uA_i \) and \( uA_{i+1} \) equal 0.5.

3. If the range of the actual data can be found by two non-zero fuzzy membership \( uA_i \) and \( uA_{i+1} \), and the range of actual data is \( \{c_i, c_{i+1}\} \), then the actual data can be obtained by the inverse operation of the fuzzy membership function and fuzzy data, which means that the defuzzification of fuzzy time series is also straightforward and simple.

The symmetric triangular fuzzy membership function is as follows.
Advances in Engineering Software 173 (2022) 103212

S. Xian et al.

range of series. Assuming metric triangular fuzzy membership function to fuzzify the training time actual forecasted data is in interval of the actual forecasted data, the actual forecasted data can be calculated by the inverse operation of Equation (13).

\[
\begin{align*}
\text{actual data} & = c_3 - 0.3 \times (c_3 - c_2) = 1700 \\
\end{align*}
\]

The main procedure of FTSOAX is illustrated in Fig. 6.

5. Application

In this section, we will give an application to illustrate the performance of FTSOAX. Recently, COVID-19 has become the center of discussion all over the world. We got the dataset that the COVID-19 daily confirmed cases in 7 countries viz. USA, India, Russia, Iran, Norway, UK, and Japan from a GitHub repository of CSSE. The COVID-19 daily confirmed cases from June.09.2020 to June.22.2021 are used to be training data in the proposed model, and test data are the COVID-19 daily confirmed cases from June.23.2021 to July.29.2021. Meanwhile, several fuzzy time series forecasting models will be compared with FTSOAX to prove that FTSOAX has better performance.

5.1. Application of forecasting the COVID-19 daily confirmed in training phase

In the training phase, we execute our model FTSOAX and other models in the COVID-19 daily confirmed in 7 countries from June.07.2020 to June.22.2021, and the range of daily confirmed is (9305. 414188). The details of the experiment are as follows. The number of intervals and the order of fuzzy time series are 7 and 2, respectively. The population and iteration of the ISOA are 10 and 50, respectively. The Powell algorithm runs once every 10 iterations. The number of iterations of XGBoost is 300. Compare FTSOAX with the following models, and we can convince that FTSOAX has better performance. Chen [39] proposed first order conventional fuzzy time series. Efendi [40] proposed first order improved weighted fuzzy time series. Sadaei [41] proposed first order exponentially weighted fuzzy time
### Table 4
Comparison between FTSOAX and other models in the training phase.

|        | Chen  | Efendi | Sadaei | Kumar | Naresh | FTSOAX |
|--------|-------|--------|--------|-------|--------|--------|
|        | RMSE  | SMAPE  | RMSE  | SMAPE | RMSE  | SMAPE  | RMSE  | SMAPE | RMSE  | SMAPE | RMSE  | SMAPE |
| USA    | 24131.75 | 26.87 | 21763.26 | 23.98 | 24518.61 | 32.76 | 18879.43 | 18.71 | 29838.7 | 36.3 | 10236.83 | 15.18 |
| India  | 15489.5 | 30.01 | 14332.2 | 29.46 | 13831.16 | 24.02 | 10915.5 | 10.17 | 27169.69 | 44.77 | 6476.96 | 7.81 |
| Russia | 2704.43 | 18.69 | 1957.36 | 11.66 | 2133.67 | 13.75 | 1786.00 | 7.80 | 2326.98 | 16.66 | 1240.49 | 3.62 |
| Iran   | 4217.34 | 34.43 | 2556.53 | 24.65 | 2671.36 | 18.4 | 2674.77 | 10.98 | 3413.13 | 33.3 | 1735.81 | 8.08 |
| Norway | 185.02 | 67.15 | 133.82 | 40.69 | 141.49 | 52.53 | 139.19 | 54.96 | 76.79 | 31.63 | 1240.49 | 3.62 |
| UK     | 4111.23 | 54.21 | 3167.46 | 43.44 | 3202.68 | 36.4 | 2797.58 | 16.99 | 7875.2 | 82.92 | 1343.01 | 14.53 |
| Japan  | 1323.14 | 57.91 | 883.2 | 50.46 | 900.65 | 40.46 | 919.81 | 24.06 | 1285.44 | 64.82 | 465.13 | 15.45 |

![Fig. 7.](attachment:image.png)

Forecasting of daily confirmed cases of India in the training phase.
S. Xian et al.

Advances in Engineering Software 173 (2022) 103212

series. Naresh [42] used a fuzzy time series model combined with particle swarm optimization to forecasting COVID-19 confirmed cases. Kumar [43] proposed a novel hybrid fuzzy time series model for the prediction of COVID-19 infected cases and deaths in India. Those models run on the dataset of COVID-19 daily confirmed cases to compare FTSOAX’s training result.

Firstly, according to the steps of FTSOAX in the previous section, we need to use ISOA to partition the universe of discourse $U$. Taking the upper and lower bounds (8635, 414188) of $U$ as input, we partition the $U$ into 7 intervals. We compare the partition results of the 50th iteration of ISOA with the results of the traditional partition method.

Converting all-time series into fuzzy time series, we can use XGBoost to predict the trend of fuzzy membership of each interval. For example, the order of fuzzy time series is 2, so we can get $F(3)_{\text{forecast}} = XGBoost(F(1), F(2))$. By analogy, we can get all forecasted fuzzy time series.

$$F(4)_{\text{forecast}} = XGBoost(F(3), F(2))$$

$$F(5)_{\text{forecast}} = XGBoost(F(4), F(3))$$

$$\vdots$$

$$F(n)_{\text{forecast}} = XGBoost(F(n), F(n-1))$$

Training XGBoost and getting all fuzzy data, the next step is the defuzzification, and its details are shown in the previous section. So we get the actual forecast data and take the RMSE of actual forecast data and original data as the fitness function of ISOA. The general flow of FTSOAX is described in Fig. 6, and the main result of FTSOAX in this application is as follows.

Aiming to illustrate the performance of FTSOAX, we give two crucial performance indicators, which reveal that FTSOAX has better performance in those indicators in the training phase. We select RMSE (Root Mean Square Error), and SMAPE (Symmetric Mean Absolute Percentage Error) as the evaluation criteria. Both of them are classic evaluation criteria. The smaller the indicators, the better. The results of all indicators are as Table 4 Comparing FTSOAX with other models in Table 4. FTSOAX’s RMSE are the best in all countries, and the second best results are behind it by [8642.60, 4438.54, 545.51, 820.72, 53.71, 1454.57, 418.07] in 7 countries. The SMAPE of FTSOAX are the best in all countries, and the second best results are behind it by [3.53, 2.36, 4.18, 2.90, 4.65, 2.45, 8.61] in 7 countries. We can explicitly uncover that FTSOAX has better performance and is superior to other models in the training phase.

We have chosen the daily confirmed cases of India as a graph to more vividly show the difference between FTSOAX and other models. The results of these models with India’s data are clearly illustrated in Fig. 7. “(a)”, “(b)”, “(c)”, “(d)”, “(e)”, and “(f)” in Fig. 7 are the results of Chen, Efendi, Sadaei, Kumar, Naresh, FTSOAX, respectively. Fig. 8 depicts the process of iteration in the FTSOAX training phase to better describe the

$U_{\text{FTSOAX}} = \{(8635, 60699), (60699, 146466), (146466, 172925), (172925, 229232), (229232, 251605), (251605, 383826), (383826, 414188)\}$

$U_{\text{tradition}} = \{(8635, 71747), (71747, 135724), (135724, 199700), (199700, 263677), (263677, 327653), (327653, 391630), (391630, 414188)\}$

$U_{\text{FTSOAX}}$ is the intervals of ISOA, and $U_{\text{tradition}}$ represents the results of the traditional partition method. The diversity of the length of intervals of $U_{\text{FTSOAX}}$ is greater than $U_{\text{tradition}}$. The traditional partition method tries to make each interval the same length, making the accuracy of the prediction results worse. Because the density of samples in the same length intervals may be different. There are even no samples in some intervals. The importance of each sample is also different. For example, the sample close to the predicted time is more important than the other samples, and the volume of each sample is difficult to express. So we use ISOA as a tool for partitioning. In this way, we do not need to consider the density and importance of the sample, and satisfactory intervals can be obtained by continuous iteration.

After partitioning the $U$ into 7 intervals, we can get the fuzzy time series with the help of the symmetric triangular fuzzy membership function. For example, Taking two samples $(x_1 = 45720, x_2 = 49310)$ as original data, the corresponding fuzzy data can be obtained. Let $C = (8635, 34667, 103583, 159696, 201079, 2404189, 317716, 399007, 414188)$ is the interval’s center, upper and lower limits, and both $x_1$ and $x_2$ are in the range $(34667, 103583)$. So The fuzzy data $F(1)$ and $F(2)$ can be calculated by Equation (13).

$$F(1) = \{0.84, 0.16, 0, 0, 0, 0, 0\}$$

$$F(2) = \{0.79, 0.21, 0, 0, 0, 0\}$$

![Fig. 8. The process of iterations of ISOA.](image-url)
Fig. 9. Forecasting of daily confirmed cases of India in the test phase.

Table 5
Comparison between FTSOAX and other models in the test phase.

|        | Chen   | Efendi | Sadaei | Kumar | Naresh | FTSOAX |
|--------|--------|--------|--------|-------|--------|--------|
| RMSE   | 29479.49 | 23865.82 | 25450.10 | 4450.10 | 4450.10 | 247.72 |
| SMAPE  | 65.89  | 52.36  | 8.61   | 8.61  | 8.61   | 7.08   |

S. Xian et al.
effect of the improved SOA algorithm. The blue and red lines in Fig. 7 are the training data and the forecasted data, respectively. Comparing the result of FTSOAX and the result of other models, it is easy to perceive that the result of FTSOAX in Fig. 7 more coincides with training data than the results of other models. By comparing the figures, we can directly find that FTSOAX has better performance in the training phase.

By comparing figures and tables, there is no doubt that FTSOAX has better performance in the training phase. To make this conclusion more persuasive, we run FTSOAX and other models on test data and compare the results of these models.

5.2. Application of forecasting the COVID-19 daily confirmed in test phase

In the test phase, we execute FTSOAX, and other models in test data of the COVID-19 daily confirmed cases from June.23.2021 to July.29.2021. Test data are forecasted by trained models in the previous subsection, and as in the training phase, we chose a dataset from India as the graph to show the results of models in Fig. 9. (a), (b), (c), (d), (e), and (f) are the results of Chen, Efendi, Sadaei, Kumar, Naresh, FTSOAX, respectively. Evaluation indicators of the test phase are shown in Table 5.

FTSOAX’s RMSE are the best in all countries except Norway, and the second best result are behind it by [1276.68, 536.68, 105.30, 200.30, 76.47, 39.46] in 6 countries. FTSOAX’s SMAPE are the best in all countries, and the second best results are behind it by [1.71, 2.01, 0.32, 3.45, 0.68, 0.66, 1.73] in 7 countries. It is explicit that the result of FTSOAX is better than any result of other models through the comparison of the figure and table. The forecasted data line of FTSOAX in Fig. 9 is more coinciding with the test data line than other models. And FTSOAX is superior to other models in most indicators. The above comparisons show that FTSOAX has an absolute advantage in the test phase. Combined with the results of the training phase and the test phase, the FTSOAX is proven to have better performance in this application.

5.3. Comparison of generalization between FTSOAX and other models

FTSOAX is compared with other models in the performance of COVID-19’s confirmed prediction in the subsections above. The results show that the performance of FTSOAX is better than other models in COVID-19’s confirmed prediction. However, the results of the application are not enough to show the advantages of FTSOAX. For this reason, in this subsection, we will further discuss the generalizability of FTSOAX and other models.

For simplicity of discussion, the Indian dataset is chosen as the training set, but the difference from the previous subsection is that the length of the test set is increased from 37 to 180 to explore the generalization of these models. The results of FTSOAX and other models on the extended test set are shown in Fig. 10 and Table 6. Fig. 10 shows the SMAPE of the models from 30 days to 180 days in the test set. Table 6 displays the RMSE and SMAPE of FTSOAX and other models in the test set over multiple periods.

The test set is divided into 3 periods of time, 40 days, 100 days, and 180 days in Table 6. The RMSE and SMAPE of these models were calculated for each period. It is clear from Table 6 that FTSOAX is superior to the other models in all periods. Further, the stability of FTSOAX is also the best among these models. Its 40 days and 180 days SMAPE differ by 3.46, which is smaller than that of other models. Fig. 10 shows the change in SMAPE of FTSOAX and other models as the length of the test set varies. Intuitively, the line indicating FTSOAX is below the other lines at all times, and it fluctuates less than the other lines. It is clear at a
glance that the accuracy and stability of FTSOAX are superior to other models in the Indian dataset. Analyzing Table 6 and Fig. 10, it is obvious that FTSOAX is generalizable and its accuracy is always better compared to other models in this case.

5.4. Comparison of robustness between FTSOAX and other models

Robustness is a very important role in evaluating the quality of a model. In this subsection, the robustness of FTSOAX is compared with other models by adding noise and outliers to the training set. As in the previous subsection, to simplify the discussion, daily confirmed of COVID-19 in India is selected as the data set.

\[ x_i = x_i + \text{factor} \times \text{Gauss}(\mu, \sigma^2) \]  

The noise is added to each element \( x_i \) of the training set, and the noise follows a Gaussian distribution with mean \( \mu \) and variance \( \sigma^2 \). There is a hyperparameter \( \text{factor} \) whose value is in the range \([0,1]\) and it controls how much noise is added. The \( \mu \) and \( \sigma^2 \) of the Gaussian distribution of the noise is 0 and the variance of the training set in the actual experiment, respectively. The training set is severely deformed, which means that the experiment is of little significance when \( \text{factor} \) is greater than 0.10. Therefore \( \text{factor} \) was decided to be 4 values \([0.01,0.02,0.05,0.10]\).

Table 7 and Table 8 present the results of the models on the training set with noise.

| factor | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE |
|--------|------|-------|------|-------|------|-------|------|-------|------|-------|
| 0.01   | 15489.50 | 30.01 | 14332.20 | 29.46 | 13831.16 | 24.02 | 10756.61 | 10.19 | 16124.82 | 34.02 |
| 0.02   | 15248.71 | 28.68 | 14642.24 | 29.59 | 14169.22 | 27.89 | 11675.76 | 11.06 | 28239.31 | 48.21 |
| 0.05   | 15133.27 | 26.95 | 14888.04 | 32.15 | 14316.33 | 21.65 | 12475.51 | 14.45 | 14633.65 | 30.61 |
| 0.10   | 16848.19 | 34.93 | 16035.76 | 34.57 | 16583.10 | 29.72 | 14119.55 | 20.63 | 18575.23 | 36.97 |

Table 8

Comparison between FTSOAX and other models in the test set with noise.

| factor | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE |
|--------|------|-------|------|-------|------|-------|------|-------|------|-------|
| 0.01   | 12333.44 | 25.21 | 13355.95 | 27.67 | 13526.97 | 22.91 | 5064.71 | 10.16 | 4335.91 | 8.66 |
| 0.02   | 10711.87 | 21.57 | 11302.56 | 22.83 | 11344.07 | 18.19 | 5548.21 | 11.13 | 4826.60 | 9.20 |
| 0.05   | 7270.70 | 14.14 | 5970.82 | 11.40 | 9978.66 | 19.97 | 6715.59 | 13.18 | 5122.42 | 9.96 |
| 0.10   | 12909.98 | 33.51 | 13438.25 | 35.40 | 7973.43 | 28.00 | 7286.11 | 15.63 | 9179.95 | 33.75 |

Fig. 11. SMAPE of FTSOAX and other models in the test set with noise.
Table 9
Comparison between FTSOAX and other models in the training set with outliers.

| M  | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE |
|----|------|-------|------|-------|------|-------|------|-------|------|-------|------|-------|
| 3  | 37757.28 | 48.62 | 19953.6 | 29.57 | 19657.61 | 25.15 | 18743.39 | 13.0 | 46710.93 | 45.52 | 12111.11 | 10.24 |
| 10 | 52549.89 | 58.08 | 31439.51 | 31.83 | 31568.35 | 28.35 | 30820.18 | 21.67 | 96610.87 | 97.13 | 20495.91 | 18.10 |
| 20 | 111165.48 | 101.93 | 44516.79 | 34.19 | 44900.91 | 35.23 | 47600.95 | 27.7 | 50624.89 | 72.07 | 33641.85 | 18.74 |

Table 10
Comparison between FTSOAX and other models in the test set with outliers.

| M  | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE | SMAPE |
|----|------|-------|------|-------|------|-------|------|-------|------|-------|------|-------|
| 3  | 45763.47 | 82.55 | 13409.37 | 38.38 | 12710.11 | 36.07 | 6109.74 | 18.11 | 22474.84 | 63.21 | 6245.33 | 16.13 |
| 10 | 54084.25 | 86.23 | 13677.95 | 38.31 | 12676.24 | 36.20 | 5076.81 | 20.62 | 116712.41 | 140.22 | 7996.81 | 19.57 |
| 20 | 117202.3 | 139.91 | 21044.85 | 53.1 | 19899.54 | 52.53 | 14654.3 | 30.14 | 63843.98 | 112.28 | 12975.74 | 25.04 |

Fig. 12. SMAPE of FTSOAX and other models in the test set with outliers.

Adding outliers to the dataset is a common way to check the robustness of a model. Outliers are added to the training set according to formula (12). \( T \) means to turn an element of the training set into \( T \) times. \( \text{random}(N, M) \) denotes a function that selects \( M \) random subscripts from 0 to \( N \). In a word, there are \( M \) random elements in the training set that become \( T \) times themselves. In actual experiments, \( T \) equal to 3 is suitable, because there is no difference in the results of models with smaller \( T \), and the results of models will become very poor with larger \( T \). \( M \) is \([3,10,20]\), indicating that 3, 10, 20 elements are randomly selected from the training set.

The perspective of the noise experiment and outlier experiment and test sets, respectively. Fig. 11 illustrates the SMAPE of the models under different factor in the test set with added noise. By comparison, it can be found that the SMAPE and RMSE of models increase with the increase of noise. FTSOAX outperforms other models in the training set and test set, judging from the tables and picture, which indicates that FTSOAX has better resistance to noise than other models in this case.

\[
x_i = T \cdot x_i \\
i = \text{random}(N, M)
\]

(14)
reveals that FTSOAX has better fault tolerance and robustness compared with other models to a certain extent in this case.

6. Conclusions

In this paper, a novel fuzzy time series forecasting model (FTSOAX) was proposed by combining ISOA and XGBoost, and it is an extension of the fuzzy time series model. It is a worthwhile attempt to apply it to COVID-19 prediction. Based on a random curve action and the Powell algorithm, we enhanced the current SOA and proposed the improved SOA (ISOA). The ISOA is used to partition the universe of discourse into suitable intervals. Furthermore, it is the first application of XGBoost on fuzzy time series to forecast the change of fuzzy membership in the literature. We compared FTSOAX and the other models with an application on COVID-19 to demonstrate that FTSOAX outperformed them. The results of the experiments reveal that FTSOAX beats other models in terms of forecasting COVID-19 daily confirmed cases. Finally, there are some issues, such as time consumption, that require more investigation and development.

CRediT authorship contribution statement

Sidong Xian: Ideas; Formulation or evolution of overarching research goals and aims; Conceptualization; Writing-review and editing; Supervision; Project administration; Funding acquisition. Kaiyuan Chen: Conceptualization; Methodology; Creation of models; Formal analysis; Data curation; Computational; Writing-Original draft preparation. Yue Cheng: Testing of existing code components; Verification.

Declaration of Competing Interest

The authors acknowledged that there is no conflict of interest in this work.

Acknowledgements

The authors express their gratitude to the Editor and the anonymous reviewers for their valuable and constructive comments. And this work was supported by the Graduate Teaching Reform Research Program of Chongqing Municipal Education Commission (No. YJG212022), Chongqing Research and Innovation Project of Graduate Students (No. CYS212326) and the Science and technology project of Chongqing market supervision and Administration Bureau (No. QCSJKJ2022023).

References

[1] Zadeh LA. Fuzzy sets. Inf Control 1965;8(3):338–53.
[2] Zeng S, Zhou J, Zhang C, Merigo JM. Intuitionistic fuzzy social network hybrid MCDM model for an assessment of digital reforms of manufacturing industry in china. Technol Forecast Soc Change 2022;176:121435.
[3] Zeng S, Zhang N, Zhang C, Su W, AlbertCarlos L. Social network multiple-criteria decision-making approach for evaluating unmanned ground delivery vehicles under the pythagorean fuzzy environment. Technol Forecast Soc Change 2022;175:121414.
[4] Song Q, Chinsom B. Fuzzy time series and its models. Fuzzy Sets Syst 1993;54(3):269–77.
[5] Chen S, Zou X, Gunawan GC. Fuzzy time series forecasting based on proportions of intervals and particle swarm optimization techniques. Inf Sci 2019;500:127–39.
[6] Pratip S. Rainfall and financial forecasting using fuzzy time series and neural networks based model. Int J Mach Learn Cybern 2018;9(3):491–506.
[7] Dong Q, Ma X. Enhanced fuzzy time series forecasting model based on hesitant differential fuzzy sets and error learning. Expert Syst Appl 2021;166:114056.
[8] Guo H, Wang L, Liu X. Dynamic time alignment kernel-based fuzzy clustering of non-equal length vector time series. Int J Mach Learn Cybern 2019;10(11):3167–79.
[9] Chen S, Dang B, Phuong H. Fuzzy time series forecasting based on optimal partitions of intervals and optimal weighting vectors. Knowled Based Syst 2017;118:204–16.
[10] Mahua B, Mall K. A novel data partitioning and rule selection technique for modeling high-order fuzzy time series. Appl Soft Comput 2018;63:87–96.
[11] Lu W, Chen X, Wijtold P, Liu X, Yang J. Using interval information granules to improve forecasting in fuzzy time series. Int J Approx Reasoning 2011;57(1):1–18.
[12] Nizam Ozogur H, Ozogur G, Orman Z. Blood glucose level prediction for diabetes based on modified fuzzy time series and particle swarm optimization. Comput Intel 2021;37(11):155–67.
[13] Bas E, Uslu VR, Yolsuc U, Ergioglu E. A modified genetic algorithm for forecasting fuzzy time series. Appl Intel 2014;44(2):453–63.
[14] Van TN. Enhanced forecasting accuracy of fuzzy time series model based on combined fuzzy c-means clustering with particle swarm optimization. Int J Comput Intel Appl 2020;19(02):2050017.
[15] Xian S, Zhang J, Xiao Y, Pang J. A novel fuzzy time series forecasting method based on the improved artificial fish swarm optimization algorithm. Soft Comput 2018;22(12):3907–17.
[16] Rashad A, Salehi S, Aliyev R. Development of fuzzy time series model for hotel occupancy forecasting. Sustainability 2019;11(3):793.
[17] Vovan T, Lebthithu T. A fuzzy time series model based on improved fuzzy function and cluster analysis problem. Commun Math Stat 2020;2020:1–16.
[18] Fan X, Wang Y, Zhang M. Network traffic forecasting model based on long-term intuitionsitic fuzzy time series. Inf Sci 2020;506:131–47.
[19] Abhishek S, Sanjay K. A modified weighted fuzzy time series model for forecasting based on two-factors logical relationship. Int J Fuzzy Syst 2019;21(5):1403–17.
[20] Cem K, Zafer DA, Cagraz VO. A new fuzzy time series method based on an ARMA-type recurrent pi-sigma artificial neural network. Soft Comput 2020;24(11):8243–52.
[21] Cheng C, Chen C. Fuzzy time series model based on weighted association rule for financial market forecasting. Expert Syst 2018;35(4):12271.
[22] Younif A, Mahmod O, Abdulhlab H. Markov weighted fuzzy time-series model based on an optimum partition method for forecasting air pollution. Int J Fuzzy Syst 2020;22(5):1468–86.
[23] Gune h, Bakir MA, Adagz CH. A novel stochastic seasonal fuzzy time series forecasting model. Int J Fuzzy Syst 2018;20(3):729–40.
[24] Dinzer NG. A new fuzzy time series model based on fuzzy c-regression model. Int J Fuzzy Syst 2018;20(6):1872–87.
[25] Zhang Y, Ou H, Wang W, Zhao J. A novel fuzzy time series forecasting model based on multiple linear regression and time series clustering. Math Probol Eng 2020;2020:29–37.
[26] Dhiman G, Kumar V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. Knovel Based Syst 2019;165:169–98.
[27] Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, et al. Xgboost: extreme gradient boosting. The 22nd ACM SIGKDD International Conference ACM 2015;1 (4):1–4.
[28] Michael P. An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comput. J 1964;7(2):155–62.
[29] Yolcu OC, Alpaslan F. Prediction of TAIEX based on hybrid fuzzy time series model with single optimization process. Appl Soft Comput 2018;66:18–33.
[30] Sarem S, Mirjali S, Lewis A. Grashopper optimisation algorithm: theory and application. Adv Eng Sofw Soft 2017;105:30–47.
[31] Zhang W, Zhang S, Zhang S. A novel method based on FTS with both GA-FCM and multifactor BPNN for stock forecasting. Soft Comput 2019;23(16):6979–94.
[32] Jiang P, Yang H, Heng J. A hybrid forecasting system based on fuzzy time series and multi-objective optimization for wind speed forecasting. Appl Energy 2019; 235:786–801.
[33] Chacon Castillo J, Segura C. Differential evolution with enhanced diversity maintenance. Optim Lett 2020;14(6):1471–90.
[34] Dang D, Friedrich T, Kotzing T, Krecja M, et al. Escaping local optima using crossover with emergent diversity. IEEE Trans Evol Comput 2017;22(3):484–97.
[35] Zhe L, Han F, Ling Q. A novel particle swarm optimisation with mutation breeding. Comm Sci 2020;32(4):333–61.
[36] Quan Y, Yin G. Analyzing convergence and rates of convergence of particle swarm optimization algorithms using stochastic approximation methods. IEEE Trans Automat Contr 2014;60(7):1760–73.
[37] Zhao B, Cang Q, Tang K, Yu V. Running time analysis of the (1+1)-EA for robust linear optimization. Theor Comput Sci 2020;843:57–72.
[38] Toint P, Cailier F. On the uniform nonsingularity of matrices of search directions and the rate of convergence in minimization algorithms. J Optim Theory Appl 1977;21(4):531–59.
[39] Chen S. Forecasting enrollments based on fuzzy time series. Fuzzy Sets Syst 1996; 81(3):311–9.
[40] Efend I, Ismail Z, Deis MM. Improved weight fuzzy time series as used in the exchange rates forecasting of our dollar to ringgit malaysia. Int J Comput Intell Appl 2011;1:20(1):135–55.
[41] Sabdel HE, Enayatifar R, Abdullah AH, Gani A. Shortterm load forecasting using a hybrid model with a refined exponentially weighted fuzzy time series and an improved harmony search. Int J Elect Power Energy Syst 2014;62:218–29.
[42] Naresh K, Susan S. Particle swarm optimization of partitions and fuzzy order for stock forecasting. Commun Math Stat 2020:1–1.