Applied Soft Computing 110 (2021) 107611

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Particle swarm optimization of partitions and fuzzy order for fuzzy time series forecasting of COVID-19

Naresh Kumar [*], Seba Susan

Department of Information Technology, Delhi Technological University, Delhi, India

ARTICLE INFO

Article history:
Received 4 April 2021
Received in revised form 3 June 2021
Accepted 7 June 2021
Available online 17 June 2021

Keywords:
COVID-19
Forecasting
Fuzzy time series
Nested PSO
Pandemic
Particle swarm optimization
ARIMA
FbProphet

ABSTRACT

Major hyperparameters which affect fuzzy time series (FTS) forecasting are the number of partitions, length of partition intervals in the universe of discourse, and the fuzzy order. There are very few studies which have considered an integrated solution to optimize all the hyperparameters. In this paper, we strive to achieve optimum values of all three hyperparameters for fuzzy time series forecasting of the COVID-19 pandemic using the Particle Swarm Optimization (PSO) algorithm. We specifically propose two techniques, namely nested FTS-PSO and exhaustive search FTS-PSO for determining the optimal interval length, as an augmentation to the FTS-PSO model that optimizes the interval length and the fuzzy order. Nested PSO has two PSO loops: (i) the inner PSO optimizes the combination of fuzzy order and boundaries of intervals for a given number of partitions defined by the outer loop, and the resultant cost is fed back to the outer PSO; (ii) the outer PSO optimizes the number of partitions to reduce the cost while meeting the defined constraint. Exhaustive search FTS-PSO also has two loops where the inner loop is similar to nested FTS-PSO while the outer loop iterates over a pre-defined search space of number of partitions. We analyze the effectiveness of the two approaches by comparing with ARIMA, FbProphet, and the state-of-the-art FTS and FTS-PSO models. We adopt COVID-19 highly affected 10 countries worldwide to perform forecasting of coronavirus confirmed cases. We consider two phases of COVID-19 spread, one from the year 2020 and another from 2021. Our study provides an analytical aspect of the COVID-19 pandemic, and aims to achieve optimal number and length of intervals along with fuzzy order for FTS forecasting of COVID-19. The results prove that the exhaustive search FTS-PSO outperformed all the methods whereas nested FTS-PSO performed moderately well.

1. Introduction

As of now, almost all the countries in the world have been affected by the novel Severe Acute Respiratory Syndrome coronavirus (SARS-CoV-2) [1]. The contagious virus is spreading at a very fast rate, and has taken lives of millions of people all over the world. The most effective ways to contain the spread of the virus are distancing, isolation, masking, sanitization, and use of scientific methods without physical contacts. Finding spread pattern and predictions using the scientific methods can be useful to framing policies and containing the outbreak. Research at large scale is going on to analyze the spread behavior of the virus based on COVID-19 data of states, countries and continents considering health, population, socioeconomic, topographic, demographic, environmental conditions etc. [2–5]. Many studies have been carried out to predict the spread behavior of the virus using various forecasting models, artificial intelligence and deep learning approaches [6–9]. Prediction of spread pattern of the virus is a very important research area to alert the medical system, governments and the society to be ready to fight against the virus.

Fluctuation is a characteristic of a time series which can be utilized to analyze the pattern, and perform the predictions. Fuzzy time series (FTS) is more descriptive compared to traditional time series because it provides semantic meaning for uncertain and fluctuating data. FTS forecasting was introduced by Song and Chissom [10–12]. They performed prediction for the enrollment of students of Alabama University using FTS. Since then, the concept has evolved by continuous value addition from researchers working in this area. The first order FTS was improved by Chen using a more computationally efficient algorithm [13]. Further, Chen also proposed a higher-order forecasting model in [14]. Generally, FTS forecasting model is divided into the following phases, taken in sequence: Defining the universe of discourse (UOD); Partitioning of the UOD; Fuzzification of the historical data; Defining fuzzy logical relationships (FLRs) and their grouping; and optionally Defuzzification, as described in [13,15]. The operation to divide UOD into unequal or equal partitions is the phase of

[*] Corresponding author.
E-mail addresses: naresh.mehra1987@gmail.com (N. Kumar), seba_406@yahoo.in (S. Susan).

https://doi.org/10.1016/j.asoc.2021.107611
1568-4946/© 2021 Elsevier B.V. All rights reserved.

Partitioning of the UOD. Major hyperparameters which affect the efficiency of FTS forecasting are the number of partitions in UOD, length of the partition intervals, and fuzzy order of the model. The existing studies have focused to optimizing one or two hyperparameters of FTS to improve the forecasting results [16–18]. FTS forecasting techniques have been used in many studies due to improved prediction results. We can take advantage of existing studies to optimize most of the hyperparameters of FTS using hybrid techniques. Coronavirus is spreading at alarming rate in the world. So, there is need of study of FTS forecasting for COVID-19 predictions. Some studies are already available in the literature to predict COVID-19 cases using FTS forecasting [19,20]. We can experiment with more hyperparameters of FTS to improve the forecasting results.

Various optimization techniques have been proposed in literature to find the optimum length of intervals in UOD, for which particle swarm optimization (PSO) [21] algorithm has gained significant attention in the recent years. Kuo et al. [22] have proposed a new hybrid model to provide forecasting solution based on first order and higher order FTS combined with PSO. Chen and Kao [23] have proposed a new FTS forecasting method for Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) forecasting. The authors have defined UOD for TAIEX using average variation of slope of two-days because it is smoother compared to the single-day variation slope. The authors have used PSO and SVM to find optimum intervals and for the classification of training data, respectively. Weighting vectors of two-factor second-order Fuzzy Logical Relationship Groups (FLRGs) and partitions of UOD have been optimized using PSO in [24] to forecast NTD/USD exchange rates and TAIEX. Chen et al. [17] have proposed FTS forecasting based on FLRGs and the proportions of partitions optimized using PSO technique. Tinh and Dieu [25] have proposed a fuzzy C-means (FCM) clustering and PSO based FTS forecasting model to improve the prediction accuracy. The aforementioned studies have focused on optimizations using PSO algorithm. Since PSO has successfully been deployed before in various fuzzy hybrid models, so we decided to use the PSO algorithm to determine the optimal solutions for our proposed FTS model.

In recent years, FTS modeling has been adopted in many forecasting studies [26–28]. Accuracy of any forecast model plays a very important role in decision making or planning. Designing a prediction model with optimized FTS hyperparameters for the COVID-19 forecasting can be useful in framing policies and containing the spread of coronavirus. So, our motivation behind this study is to design a FTS based COVID-19 forecasting model with optimized FTS hyperparameters using nested methods. In this paper, we have proposed two variants of FTS-PSO; (i) Nested FTS-PSO, and (ii) Exhaustive search FTS-PSO, to forecast fuzzy time series pertaining to COVID-19 data. We have strived to optimize the number of partitions and partition intervals in UOD along with the fuzzy order. We have used two nested loops in both the approaches to optimize the aforementioned parameters. In nested FTS-PSO, the inner PSO optimizes the combination of fuzzy order and partition intervals for a given number of partitions defined by the outer loop, and the resultant cost is fed back to the outer PSO; (ii) the outer PSO optimizes the number of partitions to reduce the cost while meeting the defined constraint. Exhaustive search FTS-PSO also has two loops where the inner loop is similar to nested FTS-PSO, while the outer loop iterates over a search space of the number of partitions, and stores the result having minimum cost. The approaches find optimum combination of number of partitions, length of partition intervals in the UOD, and the fuzzy order. We have adopted COVID-19 affected 10 countries *viz.* France, Germany, India, Iran, Italy, Russia, Spain, Turkey, UK and the US for our study. The adopted countries are highly affected in the world which has motivated us to choose these countries for the COVID-19 predictions. We have considered two timelines of COVID-19 spread for our prediction analysis. In first phase i.e. during year 2020, there was lack of knowledge and lack of containment solutions for the pandemic. In second phase i.e. during year 2021, sufficient information are available, and even vaccination also started against the disease in most of the countries. In the first phase, some of the countries have faced first and second wave of the COVID-19, whereas in the second phase, some of the countries are facing a second or third wave, or are yet to face if not taken effective measures. So, we decided to consider two timelines for separate analysis; one for the starting phase, and other for the evolved phase. We compare the proposed forecasting approaches with FTS forecasting model [13], FTS-PSO [19], FbProphet [29], and ARIMA [30] using day-level confirmed cases of COVID-19 for both the timelines. Exhaustive search FTS-PSO outperformed all the compared methods on all the adopted datasets. Nested FTS-PSO performed moderately well among the compared methods. The proposed techniques provide an optimum combination of hyperparameters, improved forecasting accuracy, and analytical base for FTS forecasting of the COVID-19 spread.

We have organized our paper as follows. Literature review is presented in Section 2. Fuzzy time series model is described in Section 3. Steps of FTS forecasting are provided in Section 4. PSO algorithm is described in Section 5. The proposed nested and exhaustive search FTS-PSO models are presented in Section 6. Experimental results and discussion are provided in Section 7. Finally, conclusions are derived in Section 8.

2. Literature review

A variety of FTS forecasting models have been proposed in the literature incorporating various features. Past twenty five years of publications related to FTS forecasting, from 1993 to 2018, have been reviewed in [15]. The authors have reviewed the existing studies by classifying the stages of FTS forecasting *viz.* (1) Defining UOD, (2) Partition of UOD, (3) Fuzzification, (4) Establishing FLRs, and (5) Defuzzification if required, into two major phases: (a) Data partitioning, (b) Prediction Phase. The authors have discussed data partitioning techniques by classifying them into a tree-level hierarchy having clustering algorithms and optimization techniques at leaf level. Yolca and Alpaslan [31] have proposed a hybrid FTS model based on synchronous evaluation of three steps of FTS forecasting. In the proposed solution, fuzzy C-means (FCM) clustering, single multiplicative neuron model (SMNM), and PSO are utilized in the FTS forecasting of TAIEX. FCM is used for fuzzification, SMNM to determine fuzzy relations, and PSO to perform the training process. In yet another study [32], a stock price forecasting framework using FTS has been presented. The authors have used an automatic clustering algorithm to define different length intervals in UOD, and an autoregressive model to determine variations in forecasted data. The framework is evaluated using TAIEX forecasting. Pattanayak et al. [33] have proposed a FTS forecasting model in which FCM is used to determine non-uniform length of intervals, SVM to determine FLRs, and autocorrelation and partial autocorrelation functions to determine the order of the model. Accuracy of the model has been evaluated on ten different time series data of TAIEX, and forecasting results have been compared with existing related studies. Panigrahi and Behera [18] have addressed two key issues in high order FTS forecasting: finding the optimum length of intervals in UOD and modeling FLRs. A modified average-based method is used to find the optimum length of intervals, and machine learning techniques, namely long short-term memory (LSTM), deep belief network (DBN), and SVM are used for modeling the FLRs. Real-time data forecasting was carried out by Nannan and Chao [26] using fuzzy cognitive map scheme. A set of information granules was used to construct the fuzzy cognitive map dynamically, where PSO is utilized to find the optimal weights of parameters, and FCM algorithm is used to adjust the cluster center according to the incoming data in real-time. Tinh [19] has proposed FTS and PSO combined model to forecast confirmed cases of coronavirus in Vietnam. The author has shown that the best performance of the FTS-PSO model is obtained with 16 partitions and 5th - order FTS for one month COVID-19 dataset from March 4, 2020 to April 7, 2020. Susan et al. [34] have proposed a single gaussian mixture based cyclone frame prediction using recent three penultimate timelapse frames. The authors have claimed that recent frames give a meaningful insight into the predictions for the next frame that make the process faster and more accurate. Alyousifi et al. [27] have proposed Markov weighted FTS model to forecast air pollution in Klang city of Malaysia. They used five types of partitioning methods; Grid partition, Huarng method, Entropy method, CMeans method, and FCM method, via two stages. In a previous work [35], which is the precursor of the current work, the authors compared ARIMA [30] and FbProphet [29] models for COVID-19 forecasting. The ARIMA model was found to outperform the FbProphet model for day-level COVID-19 cases prediction. Zheng et al. [28] have improved susceptible infected (ISI) model using hybrid approach to predict COVID-19 cases in China. They have embedded the long short-term memory (LSTM) network and the natural language processing (NLP) module into the ISI model to estimate the coronavirus development trend and infection rate. The authors have claimed that considering the control measures, increase of public awareness, and effects of prevention, the proposed model can significantly improve the prediction results as compared to the traditional epidemic models. The studies have shown new directions to optimize predictions and FTS forecasting techniques.

3. Fuzzy time series model

Conventional time series is represented by real numbers whereas FTS is represented by fuzzy sets. The concept of FTS and fuzzy sets are described in [10,11] and [36] respectively. Let the universe of discourse be $U = \{u_1, u_2, \dots, u_n\}$, and a fuzzy set A defined on U is given by

$$A = f_A(u_1)/u_1 + f_A(u_2)/u_2 + \cdots + f_A(u_n)/u_n \tag{1}$$

where f_A is the membership function (MF) of A, $f_A : U \rightarrow [0, 1]$, $f_A(u_i)$ denotes the membership degree of element u_i in the fuzzy set A for $1 \leq i \leq n$.

Let $Y(t), (t = \cdots, 0, 1, 2, \dots)$, be a subset of R, be the UOD on which fuzzy sets $f_i(t)(i = 1, 2, \dots)$ are defined, and let $F(t)$ consist of $f_1(t), f_2(t), \dots$. Then, $F(t)$ is called a fuzzy time series defined on $Y(t)$. Here, $F(t)$ is viewed as a linguistic variable and $f_i(t)$ represents possible linguistic values of $F(t)$.

If there exists a relation between $F(t)$ and $F(t-1)$ as follows: $F(t-1) \rightarrow F(t)$ then, $F(t)$ is called derived from $F(t-1)$.

If maximum degree of membership of $F(t-1)$ belongs to fuzzy set A_i and $F(t)$ belongs to fuzzy set A_j, then the fuzzy logical relationship (FLR) between $F(t-1)$ and $F(t)$ can be represented as $A_i \rightarrow A_j$; where A_i and A_j are called current state and next state of the FLR respectively, and it is a first order FLR. Similarly, if $F(t-m), \dots, F(t-2), F(t-1) \rightarrow F(t)$, then the FLR is called m-order. FLRs with the same left-hand side can be grouped together and are called as fuzzy logical relationship group (FLRG).

4. FTS forecasting steps

We study the FTS forecasting steps as described in the article [13]. The step-wise explanation about the FTS forecasting is given below.

Step 1: Defining UOD.

Assume $Y(t)$ is the given historical time series dataset. Let Dmin and Dmax be the minimum and the maximum value of $Y(t)$, respectively. Then, we can define the universe of discourse U as $[Dmin - D1, Dmax + D2]$; where, we kept value of $D1$ and $D2$ as proper positive of $0.1 \times Dmin$ and $0.1 \times Dmax$ of $Y(t)$, respectively.

Step 2: Partitioning of UOD.

In this step, U is divided into equal length intervals. Let U be divided into n equal intervals denoted by $u_1, u_2, \dots, u_n$. The intervals are defined as follows:

$$u_i = (Umin + (i-1) * L, Umin + i * L] \tag{2}$$

where $(1 \leq i \leq n)$, and the length of each interval is $L = (Umax - Umin)/n$

Step 3: Defining fuzzy sets

Each interval identified in Step 2 is defined by a linguistic variable to represent different regions in the UOD. There will be n linguistic variables for n intervals. A fuzzy set A_i is defined on each linguistic variable a shown below.

$$A_i = \frac{a_{i1}}{u_1} + \frac{a_{i2}}{u_2} + \cdots + \frac{a_{in}}{u_n} \tag{3}$$

$$a_{ij} = \begin{cases} 1 & j = i \\ 0.5 & j = i-1 \text{ or } j = i+1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $a_{ij} \in [0, 1]$, and $(1 \leq i \leq n, 1 \leq j \leq n)$, the '+' symbol denotes the set union operator, a_{ij} indicates the grade of membership of u_j in the fuzzy set A_i. Membership values of fuzzy set A_i are selected according to Eq. (4).

Step 4: Fuzzification.

In this step, each historical data is mapped to an interval which is denoted by a linguistic value. The basic rule to assign a linguistic value with respect to the corresponding fuzzy set is that each interval belongs to the highest grade of membership. Consider, $Y(t)$ is actual time series and $F(t)$ is the fuzzy time series corresponding to $Y(t)$. If we follow the maximum membership rule then the fuzzy set A_1 has the highest grade of membership in the interval u_1.

Step 5: m-order fuzzy relationships and grouping.

Using definitions from Section 3, a relationship is established such that the $F(t-m), \dots, F(t-2), F(t-1) \rightarrow F(t)$, then it is called m-order fuzzy logical relationship (FLR), where $F(t-m), \dots, F(t-2), F(t-1)$ is called the current state and $F(t)$ is called the next state. In relationships, right hand side is unique *i.e.* a linguistic value cannot appear more than once on right hand side. The relationships having the same current state can be put together and a group is formed called the fuzzy logical relationship group (FLRG).

Step 6: Defuzzification.

In this step, forecasted values are calculated using FLRGs. We have followed the defuzzification process using the rules defined in the article [13]. The principles are as follows.

(i) If there is only one fuzzy relation in FLRGs, between fuzzy sets A_i and A_r such as $A_i \rightarrow A_r$, where maximum membership of A_r belongs to interval u_r, and midpoint of u_r is m_r, then the forecasted value is m_r.

(ii) If there are more fuzzy relations in FLRGs such as $A_i \rightarrow A_{r1}$, $A_i \rightarrow A_{r2}$, …, $A_i \rightarrow A_{rp}$, where maximum membership of $A_{r1}, A_{r2}, \ldots, A_{rp}$ belong to intervals $u_{r1}, u_{r2}, \ldots, u_{rp}$ respectively, and midpoints of the intervals are $m_{r1}, m_{r2}, \ldots, m_{rp}$, then the forecasted value is $\frac{m_{r1}+m_{r2}+\cdots+m_{rp}}{p}$.

(iii) If there is no fuzzy relation in FLRGs which is defined as $A_i \rightarrow \#$, where maximum degree of membership of A_i belongs to u_i, and midpoint of u_i is m_i, then the forecasted value is m_i.

5. PSO algorithm

Particle swarm optimization (PSO) has been used for optimizing the parameters of various learning models in the past. PSO is a population-based evolutionary algorithm proposed by Kennedy and Eberhart [21,37]. It is based on fish schooling or bird flocking pattern which can search optimal or nearly optimal solution of any kind of complex problems without getting trapped into local minima. In PSO, a swarm of particles swims through n-dimensional search space of an optimization problem, where each particle denotes a potential solution. The position of $k_{th}(k = 1, 2, \ldots, P)$ particle can be represented by $X_k = [x_k^1, x_k^2, \ldots, x_k^n]$ and velocity by $V_k = [v_k^1, v_k^2, \ldots, v_k^n]$. P represents the number of particles in the swarm. Each particle moves from its current position through the search space in search of optimal solution. Each particle keeps the personal best position P_{best_k} which has been recorded so far. Position of the best particle in the swarm of all the particles is recorded as global best position P_{gbest}. Initially, all particles are initialized with random values of position in the search space. Personal best P_{best_k} of k_{th} particle and global best particle P_{gbest} are updated till the pre-defined maximum iteration (t_{max}) is reached. Velocity and position of k_{th} particle is updated using the following equations.

$$V_k^{t+1} = w^t * V_k^t + C1 * Rand() * (P_{best\_k} - X_k^t) + C2 * Rand() * (P_{gbest} - X_k^t) \tag{5}$$

$$X_k^{t+1} = X_k^t + V_k^{t+1} \tag{6}$$

$$w^t = w_{max} - \frac{t * (w_{max} - w_{min})}{t_{max}} \tag{7}$$

where w_{max} and w_{min} are pre-defined inertia weight values. w^t is the inertia weight in the tth iteration and t_{max} is the maximum iteration count. V_k^t and X_k^t are the current velocity and current position respectively of a particle k in tth iteration. V_k^t is restricted to the pre-defined range $[-Vmax, Vmax]$. $C1$ and $C2$ are cognitive and social coefficients, respectively, which are also known as acceleration coefficients. $Rand()$ is a function which randomly generates a value in the range [0,1] under uniform distribution. X_k^t is restricted within UOD range $[Umin, Umax]$. Each particle position is initialized using Eq. (8).

$$x = Umin + Rand() \times (Umax - Umin) \tag{8}$$

where Umin is lower bound and Umax is upper bounds of the universe of discourse U.

6. Proposed method

In the FTS-PSO model [19,22], PSO is used in the training phase to optimize forecasting rules and parameters. Once all the optimized rules and parameters are identified in the testing phase, we can use the model to forecast the data. In [22], partition intervals for given number of partitions are optimized using the PSO algorithm in the training phase. In [19] PSO is used to optimize the length of partition intervals for various pre-defined fuzzy orders. The use of PSO significantly improves the FTS forecasting accuracy. In our proposed work, we introduce a modified FTS-PSO algorithm, in which we optimize the number of intervals in addition to the interval length and the fuzzy order. We present two ways of optimizing the FTS hyperparameters, called nested FTS-PSO and exhaustive search FTS-PSO that are described as two separate algorithms in Sections 6.1 and 6.2, respectively. Steps of proposed modifications of FTS-PSO have been presented in Algorithms 1 and 2.

Algorithm 1 : Nested FTS-PSO

```
initialize random number of partitions $n$ from $[n_{min}, n_{max}]$ and velocity of all particles' of outer PSO which is denoted as PSO-1.
while the maximum iterations of PSO-1 is not reached do
  for particle $q$, $(1 \le q \le maxParticlesOfOuterPSO)$ do
    set number of partitions equal to $n$ which is passed by PSO-1, and initialize position and velocity of all particles' of inner PSO which is denoted as PSO-2.
    sort particles' position vector into ascending order.
    while the maximum iterations of PSO-2 is not reached do
      for particle $k$, $(1 \le k \le maxParticlesOfInnerPSO)$ do
        create intervals by using current position of particle $k$
        define linguistic values according to all intervals
        fuzzify time series data by Step 4 in Section 4
        for fuzzy-order $m$, $(1 \le m \le maxFuzzyOrder)$ do
          create $m$ order fuzzy relationships and groups by Step 5 in Section 4
          calculate forecasting values by Step 6 in Section 4
          calculate the MSE for fuzzy order $m$ for the particle $k$ based on Eq. (9)
        end for
        update the personal best $m_{best\_k}$ fuzzy order and $P_{best\_k}$ position of particle $k$ according to the calculated MSE for PSO-2.
      end for
      update the global best $m_{gbest}$ order and $P_{gbest}$ position among all the particles according to the calculated $MSE_{gbest}$ for PSO-2.
      update PSO-2 inertia weight $w_{pso2}$ according to Eq. (7)
      for particle $k$, $(1 \le k \le maxParticlesOfInnerPSO)$ do
        update particle $k$ position according to Eqs. (5) and (6)
        limit particle $k$ position using Eq. (8).
      end for
    end while
    return global best combination $(m_{gbest}, MSE_{gbest}, P_{gbest})$ to PSO-1
    update the personal best $n_{best\_q}$ number of partitions of particle $q$ in PSO-1 according to the received MSE from PSO-2.
  end for
  update the global best $n_{gbest}$ number of partitions among all the particles in PSO-1.
  update PSO-1 inertia weight $w_{pso1}$ according to Eq. (7)
  for particle $q$, $(1 \le q \le maxParticlesOfOuterPSO)$ do
    update particle $q$ number of partitions according to Eqs. (5) and (6)
    limit number of partitions $n$ for a particle $q$ within $[n_{min}, n_{max}]$ using Eq. (8).
  end for
end while
return global best combination $(n_{gbest}, m_{gbest}, P_{gbest})$
```

6.1. Nested FTS-PSO

PSO algorithm finds optimum interval lengths in UOD for a given number of partitions [19,22]. In FTS forecasting problems, we may need to determine the optimum number of partitions also, along with the optimum length of partitions, to generate better forecasting results. Length of partitions and the number of partitions are dependent variables in a FTS forecasting problem. In such a scenario, we can use nested operations to optimize the dependent variables. We can say that finding an optimum combination of number of partitions, length of partitions, and fuzzy-order is an integrated optimization problem. One solution we propose for the integrated optimization is nested FTS-PSO. This approach has two PSO loops: (i) the inner PSO optimizes the combination of fuzzy order and partition-intervals for a given number of partitions determined by the outer loop, and the resultant cost will be fed back to the outer PSO; (ii) the outer PSO optimizes the number of partitions to reduce the cost while meeting the defined constraint. All steps of nested PSO are presented in Algorithm 1. To find the optimum FTS hyperparameter using Algorithm 1, parameters of outer PSO are initialized along with the random number of partitions within a given range, as mentioned in the step-1 of the algorithm. The step-2 is the iteration loop for outer PSO. Outer PSO is responsible to optimize the number of partitions.

Algorithm 2 : Exhaustive search FTS-PSO

```
1:  initialize search space of number of partitions [$n_{min}, n_{min} + 1, \ldots, n_{max}$]
2:  for partitions $n$, ($n_{min} \leq n \leq n_{max}$) do
3:    initialize all particles' positions $X_k$, velocity $V_k$, and PSO parameters
4:    sort particles' position vector into ascending order.
5:    while the maximum iterations is not reached do
6:      for particle $k$, ($1 \leq k \leq maxParticles$) do
7:        create intervals of UOD by using particle current position
8:        define linguistic values for the intervals
9:        fuzzify time series data by Step 4 in Section 4
10:       for fuzzy-order $m$, ($1 \leq m \leq maxFuzzyOrder$) do
11:         create $m$-order fuzzy relationships and grouping by Step 5 in Section 4
12:         calculate forecasting values by Step 6 in Section 4
13:         calculate the MSE for fuzzy order $m$ for the particle $k$ based on Eq. (9)
14:       end for
15:       update the personal best $P_{best\_k}$ position and $m_{best\_k}$ fuzzy order of particle $k$ according to the calculated MSE.
16:     end for
17:     update the global best $m_{gbest}$ fuzzy order, and $P_{gbest}$ position among all the particles according to the calculated MSE.
18:     update inertia weight $w$ according to Eq. (7)
19:     for particle $k$, ($1 \leq k \leq maxParticles$) do
20:       update particle $k$ position according to Eqs. (5) and (6)
21:     end for
22:   end while
23:   update global best $n_{gbest}$ number of partitions
24: end for
25: return global best combination ($n_{gbest}, m_{gbest}, P_{gbest}$)
```

Step-3 is the loop to iterate the number of particles. The value of the number of partitions is passed to the inner PSO. UOD is randomly divided into unequal intervals using the received number of partitions as given in step-4. Inner PSO optimizes the length of the partition intervals using Eq. (6) in each iteration, and for each particle, as shown in steps 6 and 7. FTS forecasting steps are followed from step-8 to step-14 to find the optimum combination of partition intervals and the fuzzy order. Optimum result from the inner PSO in the form of best intervals, best fuzzy order, and best MSE is fed back to the outer PSO. The outer PSO updates the value of number of partitions according to Eq. (6) based on received MSE. The steps are repeated as long as the maximum iterations are not reached. Final result is produced as the optimum value of number of partitions, length of partition intervals, and fuzzy-order. Due to use of nested loops, the time complexity of the proposed FTS-PSO increases non-linearly [38]. But, to analyse all the parameters in a single run, we can use nested loops with optimum range of the loop parameters to reduce the time complexity. We have proposed a new variant of the nested FTS-PSO called exhaustive search FTS-PSO having lesser time complexity.

6.2. Exhaustive search FTS-PSO

We have proposed another approach called exhaustive search FTS-PSO to find the optimum combination of the three hyper-parameters of FTS as mentioned in Section 6.1. This approach also has two loops where the inner loop is similar to nested FTS-PSO whereas the outer loop iterates over a search space of given number of partitions, and stores the optimum result among all iterations. The approach is more efficient in term of time complexity as compared to nested FTS-PSO because it iterates over a fixed search space instead of a swarm of particles trying all ranges in nested FTS-PSO. We have provided an estimate of the time complexity of the proposed algorithms in the results section. All the steps of the exhaustive search FTS-PSO is presented in Algorithm 2.

In steps 1 and 2 of Algorithm 2, search space of the number of partitions is initialized, and iterations are defined to find the optimum value. For each value of partitions, PSO is used to find the optimum value of length of partition intervals and the fuzzy order, using the defined number of iterations and number of particles as given in steps 5 and 6 of the algorithm. In an iteration, each particle position is updated using Eq. (6), and FTS forecasting on the training data is performed using the steps from 7 to 11. Best result from the PSO is recorded in each iteration over the search space of the number of partitions. Final optimum result is generated as the optimum value of number of partitions, length of partition intervals, and fuzzy-order.

7. Experimental setup and evaluation

We have used Python 3.8 to implement the proposed approaches. We have run our experiments in Intel Core i5 processor clocked at 2.40 GHz, 8 GB RAM, and 4 GB NVIDIA GTX-1650 GPU. Evaluated datasets, performance measures, experimented parameters, and results are described in the following sub-sections.

7.1. Datasets

In our study, we have adopted 10 countries *viz.* France, Germany, India, Iran, Italy, Russia, Spain, Turkey, UK, and the US for COVID-19 confirmed cases predictions. The adopted countries are the top most affected countries by COVID-19 in the world. We have performed prediction of COVID-19 data in two phases; (i) COVID-19 phase-1 (starting phase), (ii) COVID-19 phase-2 (evolved phase). In the first phase, we have used cumulative day-level infected cases of COVID-19 of each country from April 1, 2020 to October 30, 2020. In the second phase, we have used cumulative day-level infected cases of COVID-19 of India and the US from Jan 1, 2021 to May 15, 2021 for COVID-19 predictions because these two countries are highly affected in the second phase. We have taken COVID-19 datasets from GitHub repository [39] which is maintained by the Center for Systems Science and Engineering (CSSE), USA.

Table 1
FTS prediction accuracy with increasing number of partitions for the US data.

FTS	Partitions in UOD	46	47	48	49	50	51	52	53	54	55
	MSE ($\times 10^6$)	23.476	24.330	23.108	21.903	21.964	23.143	22.007	21.973	22.889	22.005

Table 2
FTS accuracy pattern with increasing number of partitions in the UOD.

FTS	Fuzzy order	1	2	3	4	5
	MSE ($\times 10^6$)	21.964	5.028	1.273	0.351	0.299

Table 3
Prediction accuracy with number of partitions and fuzzy order.

Approach	UOD partitions	Fuzzy order (MSE$\times 10^6$)				
		1	2	3	4	5
FTS-PSO	46	18.680	4.956	0.637	0.511	0.270
	47	18.688	3.831	0.573	0.321	0.382
	48	18.393	4.976	0.700	0.339	0.328
	49	20.744	3.833	1.140	0.294	0.366
	50	20.996	3.423	0.443	0.392	0.285
	51	19.322	3.907	0.800	0.326	0.334
	52	19.555	3.130	0.677	0.278	0.287
	53	18.746	5.442	0.481	0.283	0.263
	54	18.221	3.592	0.454	0.318	0.246
	55	17.745	4.333	0.567	0.394	0.249

Table 4
Parameter values used in the experimentation.

Method	Parameter	Value
FTS	Number of partitions (COVID-19 phase-1)	{30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90}
	Number of partitions (COVID-19 phase-2)	{20, 25, 30, 35, 40, 45, 50, 55, 60}
	Fuzzy order range	{1, 2, 3, 4, 5}
PSO	Number of particles	30
	Maximum number of iterations	50
	Inertia weight [w_{min}, w_{max}]	[0.4, 0.9]
	C1, C2	2

7.2. Performance measures

We have used the statistical measures; the Mean Square Error (MSE) in training phase and Mean Absolute Percentage Error (MAPE) in testing phase to evaluate the forecasting accuracy of the experimented approaches. MSE and MAPE are defined as follows.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z_i})^2 \tag{9}$$

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{z_i - \hat{z_i}}{z_i}\right| \tag{10}$$

where z_i and $\hat{z_i}$ denote actual and predicted value, respectively, for the i_{th} instance, and N is the size of the testing set.

7.3. Experimental setup

Accuracy of FTS forecasting highly depends on the identification of optimum value of hyperparameters. We have experimented with various ranges of the number of partitions, length of the partition intervals, and fuzzy orders. Our experiment with varying the number of partitions in the UOD is shown in Table 1. We have shown MSE values of predictions with the number of partitions ranging from 46 to 55 at first order FTS of COVID-19 confirmed case of the US, during the first phase. We can observe that improvement in FTS prediction accuracy is obtained with fluctuations. So, determining an optimum value of the number of partitions is a gray area to explore. We can apply optimization algorithms like PSO to find the optimum value of the number of partitions in the UOD.

Secondly, we have experimented with high order FTS forecasting of day-level COVID-19 confirmed cases of the US during phase-1. Prediction accuracy results, denoted by MSE with the number of partitions set to 50 and fuzzy order ranging from 1 to 5, are shown in Table 2. We adopted range of fuzzy orders from [14] in our experiments. We can observe that prediction accuracy is improving with the increasing fuzzy order of the FTS of the US data, similar to the findings in the study [19].

Next, we have experimented with combination of the number of partitions and fuzzy orders of FTS using infected cases of phase-1 of COVID-19 in the US. We use PSO algorithm to optimize the length of the partition intervals in the UOD. The prediction accuracy results, in terms of MSE, for different combinations are shown in Table 3. We can observe from the results of FTS-PSO model that forecasting accuracy initially increases with increasing number of partitions and starts fluctuating after some point, and a similar trend is observed with increasing fuzzy orders. So, a best combination can be found somewhere within the given ranges. In case of FTS-PSO model, we can observe that FTS forecasting accuracy has improved due to use of PSO for interval length optimization.

From the experimental analysis, we can conclude that optimum values of the hyperparameters can be determined by experimenting with various values and using optimization techniques. But finding all the hyperparameters in an integrated way is a challenging and time-consuming task. We can say that it is an integrated optimization problem. In this study, we present the nested and exhaustive techniques to optimize the FTS hyperparameters.

We have used COVID-19 confirmed cases datasets to evaluate the performance of the proposed and comparison techniques. The COVID-19 dataset is different from other existing time series datasets because it has a continuously increasing pattern for the time period we have selected for our study. Training and testing accuracy of the presented approaches is estimated using MSE (9) and MAPE (10) functions, respectively. We have experimented with the number of partitions of UOD ranging from 30 to 90 in phase-1 and from 20 to 60 in phase-2, and fuzzy-order ranging from 1 to 5 for FTS as shown in Table 4. We selected the range of the number of UOD partitions in such a way that its maximum value shall be less than half of the data points in UOD. We have experimented with the PSO parameters as shown in Table 4 for FTS-PSO, nested FTS-PSO and exhaustive search FTS-PSO.

We have adopted optimum values of PSO parameters from literature studies, and per experiments in [19]. Selection of maximum count of iterations in our experiments is finalized after analyzing various MSE convergence patterns with respect to iteration counts. Convergence graph of accuracy (MSE) with iteration counts for training data of confirmed cases of the US is shown in Fig. 1. We can observe that accuracy is improving with the increasing number of iteration counts. MSE value converges, in this example, to an acceptable low value at iteration count 50. We observed the same MSE convergence pattern in most of the experiments with other datasets.

Further, we have analyzed the time complexity of the proposed techniques. As we have used nested loops to optimize all

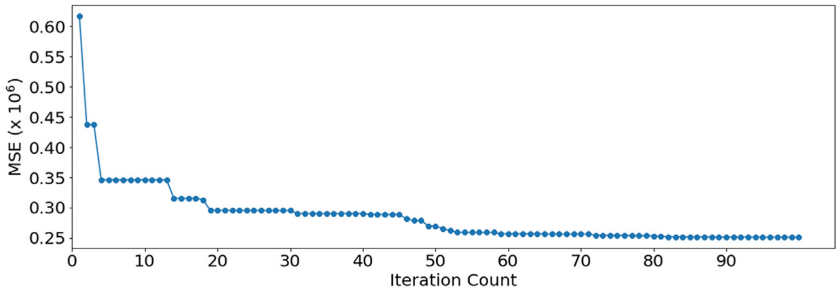


Fig. 1. Prediction accuracy (MSE) convergence with iteration counts for the US COVID-19 confirmed cases.

Table 5
Run time analysis of the proposed approaches.

Approach	Single run time (A)	Outer loop iterations (B)	Inner loop iterations (C)	Total time (A x B x C)
Nested FTS-PSO	37.869 ms	50	50	94.673 s
Exhaustive FTS-PSO	37.869 ms	13	50	24.615 s

Table 6
Best Performance of the methods for COVID-19 confirmed cases in adopted countries.

Country	Optimal hyperparameters for all methods and MAPE					
	FbProphet	ARIMA	FTS	FTS-PSO	Nested FTS-PSO	Exhaustive FTS-PSO
		(P, D, Q)	(N, O)	(N, O)	(N, O)	(N, O)
France	18.833	(5, 1, 1) 0.766	(85, 4) 0.049970	(80, 3) 0.045644	(90, 5) 0.047976	(70, 5) 0.033117
Germany	17.195	(4, 1, 1) 0.868	(80, 4) 0.028425	(85, 5) 0.016065	(90, 4) 0.020281	(70, 5) 0.014956
India	5.162	(4, 1, 1) 0.315	(85, 3) 0.005724	(90, 2) 0.003991	(90, 5) 0.003837	(90, 5) 0.003050
Iran	5.164	(3, 1, 1) 0.025	(70, 3) 0.004722	(80, 5) 0.004520	(90, 5) 0.004662	(85, 5) 0.004417
Italy	22.589	(4, 1, 1) 0.877	(60, 5) 0.025832	(90, 4) 0.022416	(89, 5) 0.024141	(75, 5) 0.021285
Russia	8.389	(5, 1, 1) 0.778	(55, 2) 0.003989	(90, 4) 0.003450	(90, 5) 0.003731	(65, 5) 0.003314
Spain	3.407	(3, 1, 1) 0.173	(90, 5) 0.009503	(55, 4) 0.008251	(90, 5) 0.007637	(80, 5) 0.006277
Turkey	0.631	(5, 1, 1) 0.043	(75, 5) 0.005271	(80, 3) 0.004107	(90, 5) 0.008281	(75, 5) 0.003275
UK	26.075	(5, 1, 1) 0.128	(85, 3) 0.006306	(90, 4) 0.005590	(90, 5) 0.005791	(90, 5) 0.005188
US	3.236	(5, 1, 1) 0.213	(50, 5) 0.003933	(75, 2) 0.003726	(90, 4) 0.003891	(85, 5) 0.002699

the hyperparameters in a single run which increases the time complexity non-linearly [38]. We have recorded run time of the techniques on a computing system having specifications as mentioned in the Section 7. We have recorded run time using number of partitions as 50, fuzzy order 1 to 5, and number of particles as 1 for phase-1 COVID-19 data of the US. We estimate the overall time based on the number of iteration used in the outer loop and the inner loop of the approaches as shown in Table 5. We can see that run time of exhaustive search FTS-PSO is very less due to lesser number of iterations in the outer loop. Predictions and comparative analysis of the study is presented in the next section.

7.4. Results and discussions

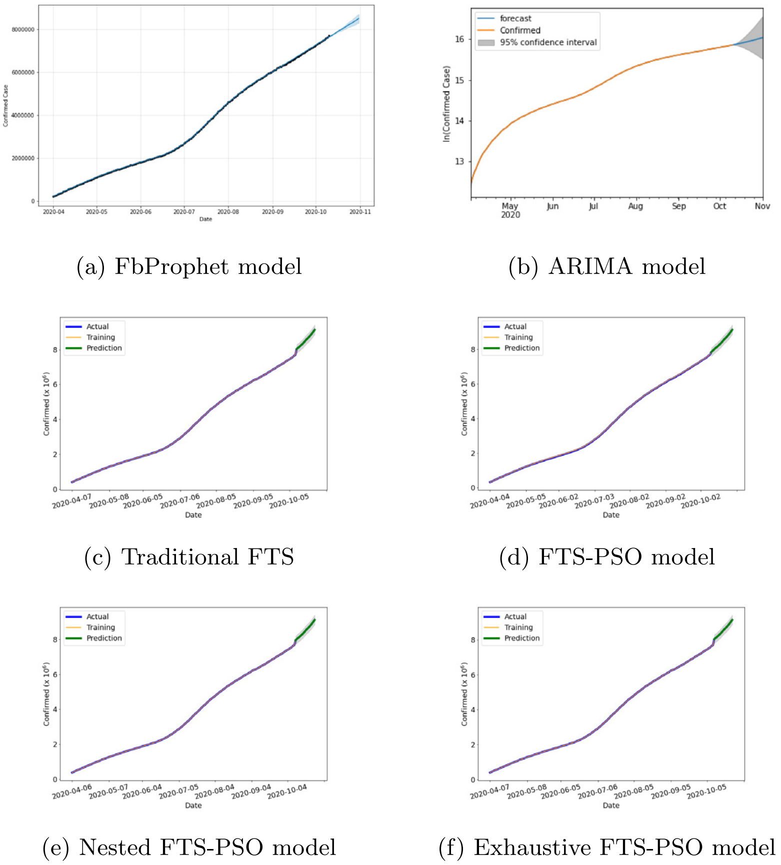
We evaluate prediction approaches *viz.* FbProphet [29], ARIMA [30], FTS [13], FTS-PSO [22], nested FTS-PSO, and exhaustive search FTS-PSO using both the timelines 7.1 of COVID-19. We show prediction results of COVID-19 day-level cumulative confirmed cases of the 10 adopted countries using the best of presented approaches. We split the datasets into training and testing samples such that the last 20 and 15 samples are used as testing samples for phase-1 and phase-2 of COVID-19 timelines, respectively. We evaluate the forecasting accuracy as per the trend followed in [28,40,41]. We have executed 10 runs of each experiment, and the best result of runs is taken as the final result.

7.4.1. Prediction results of COVID-19 for phase-1 (Starting phase)

In this section, we have shown prediction results for COVID-19 timeline from starting of April 2020 to end of October 2020 that we call as phase-1. The prediction results are generated using COVID-19 confirmed cases of the 10 adopted countries as shown in Table 6. The results are achieved for the partitions range and fuzzy order range mentioned in Table 4 for COVID-19 phase-1. The prediction accuracy for the test set is estimated using MAPE (10) function for all the compared models. One purpose of this comparison is to evaluate the presented approaches on different datasets.

In Table 6, the forecasting results from FbProhet is generated using daily seasonality. FbProphet has performed worst among the compared models. There are three major parameters of ARIMA(P, D, Q) model *viz.* order of auto regressive (P), order of differencing (D), and order of moving average (Q). Best performed (P, D, Q) parameters and respective prediction accuracy (MAPE) of ARIMA are shown in Table 6 for each country. It has performed better than FbProphet, which is in agreement with the conclusions in our previous work [35]. There are two major parameters to be initialized in FTS forecasting *viz.* number of partitions (N) and fuzzy-order (O). Best performing pair of (N, O) which has yielded the best accuracy (MAPE) is shown in the table for FTS, FTS-PSO, nested FTS-PSO, and exhaustive search FTS-PSO. Optimization of the length of intervals for each pair is performed using PSO in case of FTS-PSO and purposed techniques. Effect of PSO can be seen by comparing the results of FTS and FTS-PSO in Table 6. Accuracy has been improved significantly by using PSO with FTS. We can see that nested FTS-PSO has performed moderately well as compared to other approaches. We observe that it was subject to overfitting during the training phase. The approach consistently followed the same pattern for all the datasets. A stop condition is required before actual predictions in case of nested FTS-PSO. However the approach is able to generate competitive results. We can see that exhaustive search FTS-PSO has outperformed all the compared approaches in case of all the adopted countries. It is able to adopt the scenarios and generate the best forecasting results.

We have shown forecasting results of all the approaches on the US dataset of phase-1 in Fig. 2. Training of the data is shown using line plot and forecasting result is depicted with shaded area of one percentage variation of the results. We can see that all models are well fitted during training period but forecasting results



(a) FbProphet model (b) ARIMA model

(c) Traditional FTS (d) FTS-PSO model

(e) Nested FTS-PSO model (f) Exhaustive FTS-PSO model

Fig. 2. Forecasting of COVID-19 confirmed cases of US for Phase-1.

Table 7
Prediction accuracy of the approaches for phase-2 timeline of COVID-19.

Country	Optimal hyperparameters for all methods and MAPE					
	Prophet	ARIMA (P, D, Q)	FTS (N, O)	FTS-PSO (N, O)	Nested FTS-PSO (N, O)	Exhaustive FTS-PSO (N, O)
India	7.411	(0, 1, 1) 0.560	(60, 4) 0.009233	(60, 2) 0.008289	(60, 5) 0.008137	(50, 5) 0.005865
US	0.698	(5, 1, 1) 0.022	(50, 3) 0.003580	(60, 4) 0.002961	(60, 5) 0.003192	(60, 5) 0.002410

have minor variations, and measured performance in numbers is shown in Table 6 for better visualization.

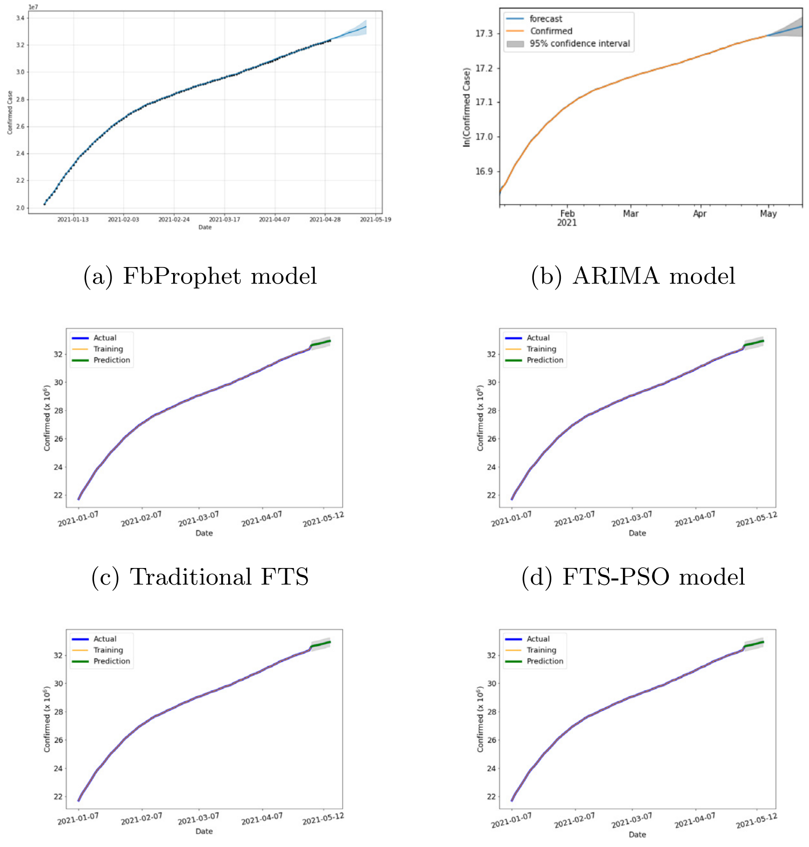
7.4.2. Prediction results of COVID-19 for phase-2 (Evolved phase)

Most of the countries have faced or are facing the second or third wave of COVID-19. So, we have preformed comparative prediction analysis of the adopted and proposed approaches on the latest data. In this prediction analysis, we have considered only India and the US because these countries were impacted most in the phase-2 timeline. We have shown the prediction accuracy results of all the approaches in Table 7 for both the countries. From the table, we can see that again exhaustive search FTS-PSO outperformed all the approaches on phase-2 timeline data of COVID-19. The pattern of performance is similar to the results in Table 6.

Further, we have depicted the performance of all the approaches on phase-2 timeline COVID-19 data of the US in Fig. 3. We can visualize the forecasting results of all the approaches in the figure. The approaches have followed the same pattern of results on the phase-2 timeline data as shown for phase-1 timeline data of the US.

8. Conclusion

Hyperparameters of FTS forecasting model are the number of partitions in UOD, partition intervals, and fuzzy order. Prediction accuracy of any FTS forecasting model depends on the tuning of its hyperparameters. In the literature, existing approaches have provided solutions for FTS forecasting problems by optimizing the interval lengths and the fuzzy order, but set the number of intervals to a pre-determined value. In this paper, we have tried to optimize the above-mentioned three hyperparameters using PSO algorithm. We have proposed nested FTS-PSO, and exhaustive search FTS-PSO models for fuzzy time series forecasting that determine the number of partitions, the length of partition intervals, and the fuzzy order. We have adopted COVID-19 dataset of confirmed cases in highly affected 10 countries. We have compared the performance of our models with the ARIMA, FbProphet, conventional FTS and FTS-PSO models. We have presented prediction results of all the approaches on two timelines i.e. starting and evolved phases of COVID-19. The proposed approaches are able to find out the optimum combination of hyperparameters. The proposed exhaustive search FTS-PSO has outperformed all the compared approaches on the both the timelines data. We can further improve the analysis of COVID-19 predictions by incorporating lockdown effects and government policies.



(a) FbProphet model

(b) ARIMA model

(c) Traditional FTS

(d) FTS-PSO model

(e) Nested FTS-PSO model

(f) Exhaustive FTS-PSO model

Fig. 3. Forecasting of COVID-19 confirmed cases of US for phase 2.

CRediT authorship contribution statement

Naresh Kumar: Conceptualization, Methodology, Software, Data curation, Writing- original draft, Visualization, Investigation. **Seba Susan:** Conceptualization, Methodology, Investigation, Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Sohrabi, Z. Alsafi, N. O'Neill, M. Khan, A. Kerwan, A. Al-Jabir, C. Iosifidis, R. Agha, World health organization declares global emergency: A review of the 2019 novel coronavirus (covid-19), Int. J. Surg. 76 (2020) 71–76.

[2] S. Chakraborti, A. Maiti, S. Pramanik, S. Sannigrahi, F. Pilla, A. Banerjee, D.N. Das, Evaluating the plausible application of advanced machine learnings in exploring determinant factors of present pandemic: A case for continent specific covid-19 analysis, Sci. Total Environ. 765 (2021) 142723.

[3] Á. Briz-Redón, Á. Serrano-Aroca, A spatio-temporal analysis for exploring the effect of temperature on covid-19 early evolution in spain, Sci. Total Environ. 728 (2020) 138811.

[4] A. Mollalo, B. Vahedi, K.M. Rivera, Gis-based spatial modeling of covid-19 incidence rate in the continental united states, Sci. Total Environ. 728 (2020) 138884.

[5] G.F. Ficetola, D. Rubolini, Containment measures limit environmental effects on covid-19 early outbreak dynamics, Sci. Total Environ. 761 (2021) 144432.

[6] Y. Hao, T. Xu, H. Hu, P. Wang, Y. Bai, Prediction and analysis of corona virus disease 2019, PLoS One 15 (10) (2020) e0239960.

[7] X. Zhang, R. Ma, L. Wang, Predicting turning point, duration and attack rate of covid-19 outbreaks in major western countries, Chaos Solitons Fractals (2020) 109829.

[8] S. Lalmuanawma, J. Hussain, L. Chhakchhuak, Applications of machine learning and artificial intelligence for covid-19 (sars-cov-2) pandemic: A review, Chaos Solitons Fractals (2020) 110059.

[9] T.B. Alakus, I. Turkoglu, Comparison of deep learning approaches to predict covid-19 infection, Chaos Solitons Fractals 140 (2020) 110120.

[10] Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series—part i, Fuzzy Sets and Systems 54 (1) (1993) 1–9.

[11] Q. Song, B.S. Chissom, Fuzzy time series and its models, Fuzzy Sets and Systems 54 (3) (1993) 269–277.

[12] Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series-part ii, Fuzzy Sets and Systems 62 (1) (1994) 1–8.

[13] S.-M. Chen, Forecasting enrollments based on fuzzy time series, Fuzzy Sets and Systems 81 (3) (1996) 311–319.

[14] S.-M. Chen, Forecasting enrollments based on high-order fuzzy time series, Cybern. Syst. 33 (1) (2002) 1–16.

[15] M. Bose, K. Mali, Designing fuzzy time series forecasting models: A survey, Internat. J. Approx. Reason. 111 (2019) 78–99.

[16] C.A. Severiano, P.C. Silva, H.J. Sadaei, F.G. Guimarães, Very short-term solar forecasting using fuzzy time series, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–6.

[17] S.-M. Chen, X.-Y. Zou, G.C. Gunawan, Fuzzy time series forecasting based on proportions of intervals and particle swarm optimization techniques, Inform. Sci. 500 (2019) 127–139.

[18] S. Panigrahi, H.S. Behera, A study on leading machine learning techniques for high order fuzzy time series forecasting, Eng. Appl. Artif. Intell. 87 (2020) 103245.

[19] N. Van Tinh, Forecasting of covid-19 confirmed cases in vietnam using fuzzy time series model combined with particle swarm optimization, Comput. Res. Progr. Appl. Sci. Eng. 6 (2) (2020) 114–120.

[20] C. Fatih, A. Hamimes, P. Mishra, A note on covid-19 statistics, strange trend and forecasting of total cases in the most infected african countries: An arima and fuzzy time series approaches, Afr. J. Appl. Stat. 2 (2) (2020) 961–975.

[21] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[22] I.-H. Kuo, S.-J. Horng, T.-W. Kao, T.-L. Lin, C.-L. Lee, Y. Pan, An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization, Expert Syst. Appl. 36 (3) (2009) 6108–6117.

[23] S.-M. Chen, P.-Y. Kao, Taiex forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines, Inform. Sci. 247 (2013) 62–71.

[24] S.-M. Chen, B.D.H. Phuong, Fuzzy time series forecasting based on optimal partitions of intervals and optimal weighting vectors, Knowl.-Based Syst. 118 (2017) 204–216.

[25] N. Van Tinh, N.C. Dieu, A new hybrid fuzzy time series forecasting model based on combining fuzzy c-means clustering and particle swam optimization, J. Comput. Sci. Cybern. 35 (3) (2019) 267–292.

[26] Z. Nannan, L. Chao, Adaptive online time series prediction based on a novel dynamic fuzzy cognitive map, J. Intell. Fuzzy Systems 36 (6) (2019) 5291–5303.

[27] Y. Alyousifi, M. Othman, I. Faye, R. Sokkalingam, P.C. Silva, Markov weighted fuzzy time-series model based on an optimum partition method for forecasting air pollution, Int. J. Fuzzy Syst. 22 (2020) 1468–1486.

[28] N. Zheng, S. Du, J. Wang, H. Zhang, W. Cui, Z. Kang, T. Yang, B. Lou, Y. Chi, H. Long, et al., Predicting covid-19 in china using hybrid ai model, IEEE Trans. Cybern. 50 (7) (2020) 2891–2904.

[29] S.J. Taylor, B. Letham, Forecasting at scale, Amer. Statist. 72 (1) (2018) 37–45.

[30] S.S. Wulff, Time series analysis: Forecasting and control, J. Qual. Technol. 49 (4) (2017) 418.

[31] O.C. Yolcu, F. Alpaslan, Prediction of taiex based on hybrid fuzzy time series model with single optimization process, Appl. Soft Comput. 66 (2018) 18–33.

[32] W. Wang, A big data framework for stock price forecasting using fuzzy time series, Multimedia Tools Appl. 77 (8) (2018) 10123–10134.

[33] R.M. Pattanayak, S. Panigrahi, H. Behera, High-order fuzzy time series forecasting by using membership values along with data and support vector machine, Arab. J. Sci. Eng. 45 (12) (2020) 10311–10325.

[34] S. Susan, A. Saxena, A. Budhwar, A. Takhi, A. Varshney, Cyclone frame prediction by gaussian mixture modeling of the three penultimate time-lapse frames, J. Indian Soc. Remote Sens. 45 (5) (2017) 899–901.

[35] N. Kumar, S. Susan, Covid-19 pandemic prediction using time series forecasting models, in: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2020, pp. 1–7.

[36] L. Zadeh, Fuzzy sets information and control, MathSciNet zbMATH 8 (1965) 338–353.

[37] J. Kennedy, Swarm intelligence, in: Handbook of Nature-Inspired and Innovative Computing, Springer, 2006, pp. 187–219.

[38] L. Shi, S. Olafsson, Nested partitions method for global optimization, Oper. Res. 48 (3) (2000) 390–407.

[39] G. Inc., Covid-19 cases, 2020, https://github.com/cssegis{and}data/covid-19 (accessed in november 1, 2020).

[40] A. Zeroual, F. Harrou, A. Dairi, Y. Sun, Deep learning methods for forecasting covid-19 time-series data: A comparative study, Chaos Solitons Fractals 140 (2020) 110121.

[41] F. Rustam, A.A. Reshi, A. Mehmood, S. Ullah, B.-W. On, W. Aslam, G.S. Choi, Covid-19 future forecasting using supervised machine learning models, IEEE Access 8 (2020) 101489–101499.

Naresh Kumar received the B.Tech degree in Computer Engineering from Malviya National Institute of Technology, Jaipur, India in 2009, and the M.Tech degree in Computer Science and Automation from Indian Institute of Science, Bengluru, India in 2012. He joined Central Research Laboratory, Bharat Electronics Ltd as research staff member in 2012. He is currently pursuing the Ph.D. degree in the Department of Information Technology at Delhi Technological University (DTU), India. His research interests are in the area of Time series forecasting, Soft Computing, and Routing in mobile ad hoc networks. His current research involves prediction of the Covid-19 pandemic from time-series data.

Seba Susan is a Professor in the Department of Information Technology at Delhi Technological University (DTU). She completed her Ph.D from the Electrical Engineering Department of Indian Institute of Technology (IIT) Delhi in 2014. Her research areas are Soft Computing, Data Mining, Computer Vision, Speech and Natural Language Processing.