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.

Applied Soft Computing 111 (2021) 107683

Contents lists available at ScienceDirect

Applied Soft Computing

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

Transmission trend of the COVID-19 pandemic predicted by dendritic neural regression

Minhui Dong [a], Cheng Tang [b], Junkai Ji [a,*], Qiuzhen Lin [a], Ka-Chun Wong [c]

[a] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China
[b] Faculty of Engineering, University of Toyama, Toyama-shi, 930-8555, Japan
[c] Department of Computer Science, City University of Hong Kong, Hong Kong, China

ARTICLE INFO

Article history:
Received 2 February 2021
Received in revised form 9 June 2021
Accepted 28 June 2021
Available online 7 July 2021

Keywords:
COVID-19
Prediction
Regression
Neural network
Optimization

ABSTRACT

In 2020, a novel coronavirus disease became a global problem. The disease was called COVID-19, as the first patient was diagnosed in December 2019. The disease spread around the world quickly due to its powerful viral ability. To date, the spread of COVID-19 has been relatively mild in China due to timely control measures. However, in other countries, the pandemic remains severe, and COVID-19 protection and control policies are urgently needed, which has motivated this research. Since the outbreak of the pandemic, many researchers have hoped to identify the mechanism of COVID-19 transmission and predict its spread by using machine learning (ML) methods to supply meaningful reference information to decision-makers in various countries. Since the historical data of COVID-19 is time series data, most researchers have adopted recurrent neural networks (RNNs), which can capture time information, for this problem. However, even with a state-of-the-art RNN, it is still difficult to perfectly capture the temporal information and nonlinear characteristics from the historical data of COVID-19. Therefore, in this study, we develop a novel dendritic neural regression (DNR) method to improve prediction performance. In the DNR, the multiplication operator is used to capture the nonlinear relationships between input feature signals in the dendrite layer. Considering the complex and large landscape of DNR's weight space, a new scale-free state-of-matter search (SFSMS) algorithm is proposed to optimize the DNR, which combines the state-of-matter search algorithm with a scale-free local search. The SFSMS achieves a better global search ability and thus can effectively reduce the possibility of falling into local minima. In addition, according to Takens's theorem, phase space reconstruction techniques are used to discover the information hidden in the high-dimensional space of COVID-19 data, which further improves the precision of prediction. The experimental results suggest that the proposed method is more competitive in solving this problem than other prevailing methods.

1. Introduction

The novel coronavirus disease is a global problem. Because the first patient was diagnosed in December 2019, the disease has been called COVID-19. Due to the irregularity of viral transmission that occurred during the initial stage of the COVID-19 pandemic, information on this novel coronavirus and medical facilities were insufficient; therefore, the pandemic was initially not well controlled [1]. On January 20, 2020, it was officially announced that COVID-19 could be passed from person to person [2]. Soon after, COVID-19 spread around the world. The new disease was defined as a Public Health Emergency of International Concern by the World Health Organization (WHO) on January 30, 2020 [3,4].

COVID-19 itself is not a large threat, but COVID-19 patients often have other complex and severe symptoms that can lead to death [5]. As of November 1, 2020, there were approximately 46 million diagnosed patients and 1.2 million deaths worldwide [6]. From the COVID-19 Data Hub [7], a COVID-19 open research dataset, we learned that the COVID-19 pandemic has been controlled in some areas, such as China, Burkina Faso, and Comoros, but is still severe in many other countries. In addition, even in areas where the outbreak is contained, there is still a possibility of re-outbreaks due to factors such as colder temperatures. As a result, countries still have a substantial need for COVID-19 prevention and control strategies. Good decisions can help reduce the spread of an pandemic, improve survival rates and reduce mortality. It is imperative to study the transmission trend of the COVID-19 pandemic and identify the mechanism of transmission.

For the above reasons, this paper aims to study the historical transmission trend of the COVID-19 pandemic in various countries and determine the transmission mechanism of COVID-19 to

* Corresponding author.
E-mail address: jijunkai@szu.edu.cn (J. Ji).

https://doi.org/10.1016/j.asoc.2021.107683
1568-4946/

provide important reference information for decision-makers in every country. When pandemics and major public health emergencies occur, pandemic models are usually built to analyse and predict the development trend of the disease. There are two main methods for building pandemic models. The first one is the mathematical method. There have been practical cases in which mathematical models have been applied to epidemiology in the past [8]. For example, Sharomi utilized a regression model to forecast the spread trend of tuberculosis [9], and in [10,11], a transmission model was built for malaria. A model that takes susceptible, infectious, and recovered individuals as factors (SIRS) was improved and applied to study the syncytial virus in infants in [12], and in [13], mathematical models were adopted to control the dengue outbreak.

To more effectively detect the transmission pattern of the new disease and obtain more information to help develop prevention strategies after the WHO declared the emergency, various mathematical methods have been used at varying degrees to model COVID-19 transmission. In [14–17], researchers tried to use statistical or mathematical models to analyse and predict COVID-19 transmission trends. However, the fitting results of these methods are relatively poor, and the accuracy is low. The main reason is that statistical or mathematical modelling methods are mostly linear, whereas the transmission trend of the COVID-19 pandemic is a nonlinear regression problem with temporal factors. Some researchers have taken these features into account and proposed improved mathematical models that can be applied to nonlinear problems [17–20]. However, most of these models need to impose a certain environmental assumption on the data, which may cause the loss of some important information. These methods also fail to consider temporal characteristics and are thus unable to achieve satisfactory results.

To overcome the shortcomings of mathematical methods, researchers began to adopt machine learning (ML) technology to forecast the transmission trend of the COVID-19 pandemic. According to previous studies, the powerful ability of a recurrent neural network (RNN) [21] to describe the temporal and nonlinear characteristics of data can be effectively used to predict the spread trend of COVID-19 and achieve high-level accuracy. Therefore, state-of-the-art RNNs, including long short-term memory (LSTM) [22], bidirectional long short-term memory (BiLSTM) and gated recurrent units (GRUs), have been widely employed to predict the transmission trend. In addition, a hybrid method that combined LSTM and natural language processing (NLP) was proposed to predict the COVID-19 transmission trend in China [23]. In [24], LSTM was utilized by Chimmula to estimate the endpoint of the Canadian COVID-19 pandemic. An improved convolution LSTM model was adopted by Shastri [25] to perform spread trend prediction for India and the United States and achieved excellent results. Other researchers have used RNNs to predict the transmission trend of the COVID-19 pandemic on other data sets [26–29]. Although RNNs can effectively extract temporal and nonlinear features from data, they still have some disadvantages; for instance, the temporal information cannot be fully extracted from the data by only applying a simple RNN model. RNNs also suffer from overfitting issues and unstable performance caused by different random weight initializations [30].

To further improve the precision of prediction, a modified dendritic neuron model (DNM) is developed and applied to forecast the transmission trend of the COVID-19 pandemic in this study. The computation of DNM mimics the process of biological neurons transferring information. Due to the nonlinear characteristics of synapses and the plasticity of dendrites, the DNM has a strong ability to fit complex nonlinear functions. Previous studies have used the DNM to solve multifarious linearly nonseparable problems and achieved satisfactory results [31–33]. However, the original DNM was specifically designed for classification problems. The architecture is simplified to seek high classification speed by neglecting the thickness of dendritic branches, which will influence the signal strengths [34]. This study proposes dendritic neural regression (DNR), which can enhance model regression ability by considering the thickness of dendritic branches.

Moreover, since DNR's weight has a complex and large search space, a novel scale-free state-of-matter search (SFSMS) algorithm that combines the state-of-matter search (SMS) algorithm [35] with the scale-free local search method is utilized to optimize the neural architecture of DNR. The SMS algorithm is a recently proposed evolutionary algorithm that has powerful search capabilities and can effectively avoid local optima. The scale-free method searches in a paradigm of complex networks, which are similar to a variety of real-world networks. In SFSMS, the scale-free local search is a new component that improves the quality of the solutions.

Additionally, Takens's phase space reconstruction (PSR) theorem [36] is employed as a module to preprocess the raw data in this study. The original one-dimensional time series hides some information in the high-dimensional space, which leads to intermittent behaviour and randomness. This information can be extracted through PSR so that we can greatly improve the prediction performance. Two components are imperative when we employ PSR techniques to describe high-dimensional information: the time delay and the embedding dimension. The details of how to obtain these two components are described in Section 3. Experiments for predicting COVID-19 transmission trends in several countries show that DNR-SFSMS is more accurate and robust than other commonly used ML algorithms.

The remainder of this article is organized as follows. Section 2 introduces the structure of DNR and how to use the SFSMS algorithm to optimize DNR. The whole process of data preprocessing is introduced in Section 3. Section 4 presents the experimental results and discussion. Finally, a conclusion is provided in Section 5.

2. Model structure and optimization algorithm

In this study, we employed DNR to forecast the transmission trend of the COVID-19 pandemic in several countries. The left part of Fig. 1 shows the structure of DNR. DNR consists of four layers: the synaptic layer, the dendrite layer, the membrane layer and the soma layer. The synaptic layer is the entry point of the model and is used to receive input signals. The signal received by the synaptic layer is processed by an activation function and then flows to all branches of the dendrite layer. Each branch of the dendrite layer gathers all the signals in the branch and sends them to the membrane layer. The membrane receives signals from all branches of the dendrite layer and integrates them to transmit to the soma layer. Finally, the soma layer processes the signal through a sigmoid function and outputs it.

2.1. Synaptic layer

The synaptic layer mimics the synaptic part of the nervous system; it is the portal of the neuron and receives signals from external inputs. The signals received by the synaptic layer are processed by the following equation:

$$D_{ij} = \frac{1}{1 + e^{-\alpha(w_{ij}x_i - \theta_{ij})}}, \tag{1}$$

where x_i is the ith input signal and D_{ij} denotes the value of the ith synapse transfer to the jth dendritic branch. α is a positive constant parameter in the synaptic layer. w_{im} and θ_{im} are two alterable parameters in terms of different tasks.

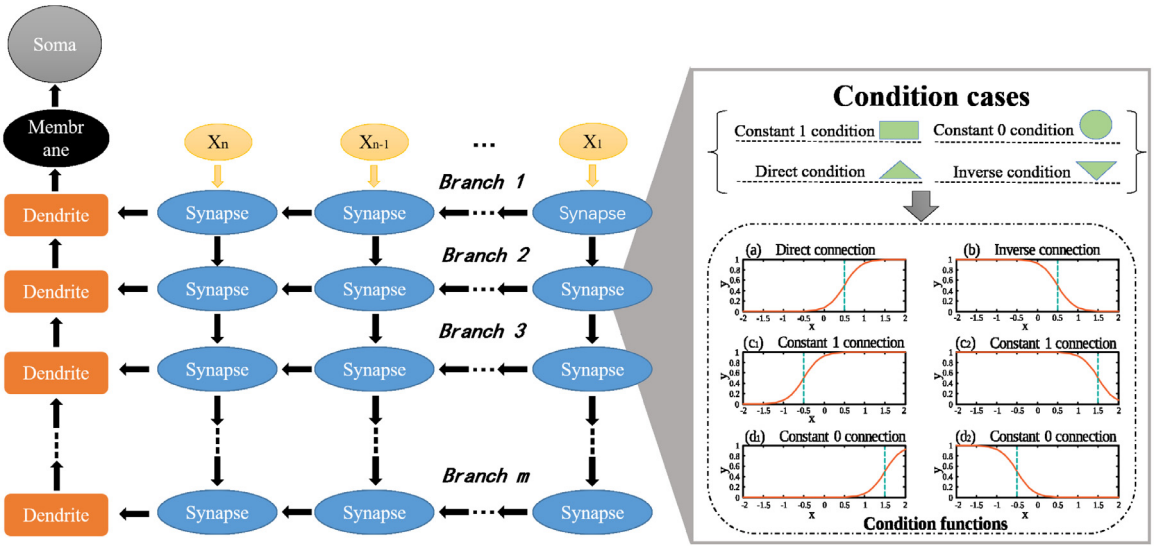

Fig. 1. Description of the DNR architecture.

Depending on the different values of w_{im} and θ_{im}, DNR can be further simplified. This operation mimics synapses that can become excitatory synapses or inhibitory synapses in terms of the received ions in the nervous system [32]. The detailed state changing rule is given as follows: (1) Constant 1 connection ($w_{ij} < 0 < \theta_{im}$ or $0 < w_{ij} < \theta_{im}$): the value of the synaptic layer output to the dendritic layer is fixed at 1; (2) Constant 0 connection ($w_{ij} < \theta_{im} < 0$ or $\theta_{im} < 0 < w_{ij}$): the value of the synaptic layer output to the dendritic layer is fixed at 0; (3) Excitatory connection ($0 < \theta_{im} < w_{ij}$): the output of the synaptic layer remains the original value; (4) Inhibitory connection ($w_{ij} < \theta_{im} < 0$): the output of the synaptic layer is the inverse of the original value.

2.2. Dendrite layer

The dendrite layer is responsible for aggregating signals from synapses distributed on each branch. The nonlinear relationship among these signals is thought to play an important role in neural information processing for some sensory systems, such as the visual and auditory systems, in biological networks [37,38]. The nonlinear relationship is described by multiplication operations in DNR, which can be expressed as follows:

$$M_j = \prod_{i=1}^{I} y_{ij}, \tag{2}$$

where M_j denotes the output value of the mth dendritic branch.

2.3. Membrane layer

The duty of the membrane layer is to integrate the signals from all branches of the dendrite. The integrated operation is implemented through a summation, which can be given as follows:

$$S = \sum_{j=1}^{J} (\mu_j * M_j), \tag{3}$$

where μ_j denotes the strength of each dendritic branch and S represents the input to the soma layer. μ_j is a parameter that differentiates DNR from DNM. In DNM, since μ_j is fixed at 1, further simplification can be performed to obtain a faster calculation speed [39]. In DNR, μ_m is a variable parameter, and it can adapt to different tasks by constantly changing its value to better cope with regression problems.

2.4. Soma layer

A sigmoid function is employed as the activated function in the soma layer, and the cell body is fired when the potential from the membrane exceeds the threshold. The process can be defined as follows:

$$R = \frac{1}{1 + e^{-\alpha(S-v)}}, \tag{4}$$

where R denotes the output of the soma layer and α and v are two positive constants.

2.5. Learning algorithm

Since DNR captures the nonlinear relationship among features by means of the multiplication operation in the dendrite layer, the parameter space is tremendous and complex. In addition, in DNR, we add additional weights to describe the thickness of the dendrite branches, which further increases the difficulty of model optimization. The search abilities of traditional back propagation (BP) algorithms have some limitations in such a parameter space, including a tendency to fall into local optima and sensitivity of the initialized weights. Thus, the SFSMS algorithm is utilized to optimize DNR instead. The SFSMS algorithm is composed of an SMS algorithm and a scale-free local search approach, where the scale-free local search is used as an additional strategy to improve the whole population in the process of evolution. In the rest of this section, the whole process of the SFSMS algorithm will be introduced specifically.

2.5.1. SMS algorithm

In nature, substances vary among gases, liquids and solids in terms of temperature. The SMS algorithm refers to this principle [35]. The process of the SMS algorithm can be regarded as a process of substantial change from gas to liquid to solid. First, in the gas state, the molecules are long distances from each other and have weak attractions, but they have a large space of motion and easily collide. When the distance between the molecules is sufficiently reduced, the substance changes from the gas state to the liquid state. In the liquid state, the attraction among the molecules increases compared with the gas state, and the space of motion of the molecules and the possibility of collision are reduced. When the distance between molecules is very small, a substance becomes solid. The attraction among the molecules in

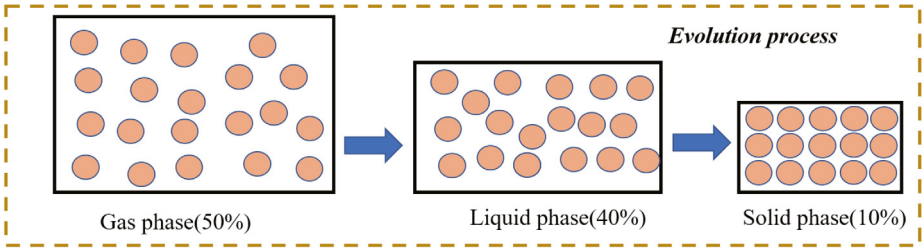

Fig. 2. Evolution process of the SMS algorithm.

the solid state is close to the maximum, the motion space is very small, and there are almost no collisions between molecules.

There are three major phases in the process of searching for the best solution, as shown in Fig. 2; the three operations include the direction vector operation, the collision operation and random behaviour. Their specific definitions are shown below.

Direction vector operation: Similar to the molecules in a substance, in SMS algorithms, the current best individual attracts other individuals in the population to raise the level of the entire population. The purpose of the direction vector operation is to move the other individuals towards the best one. Let P_i be the ith individual vector and $\bar{p}$ be the current optimal individual. According to the rules of the SMS algorithm, other individuals are biased towards $\bar{P}$ when moving. Let d_i be the direction vector of the ith individual; its formula can be expressed as follows:

$$d_i^{m+1} = d_i^m * (1 - \frac{t}{Ita}) * 0.5 + \frac{\bar{P} - P_i}{\|\bar{P} - P_i\|}, \tag{5}$$

where Ita and m denote the maximum iteration number and the current number of evolutionary iterations, respectively. Then, we can calculate the velocity vector in terms of D_i. The equation is defined as follows:

$$v_i = d_i \frac{\sum_{j=1}^{n}(b_j^{high} - b_j^{low})}{n} * \eta, \tag{6}$$

where η is a positive constant and n is the dimension of the individual vectors. b_j^{high} and b_j^{low} represent the upper bound and the lower bound of the jth member of individual vectors, respectively. The last step of this operation is to update the individual vectors, which is given as follows:

$$P_{i,j}^{m+1} = p_{i,j}^m + v_j * R_1 * (b_j^{high} - b_j^{low}) * \rho, \tag{7}$$

where R_1 is a random number and ρ is a positive constant. Both are in the range of [0,1].

Collision operation: The collision operation aims to solve the problem of population diversity loss and premature convergence in the evolution process. In this operation, a threshold is employed to determine whether two individuals have collided. When the distance between two individuals is less than the threshold, they are considered to have collided. The direction vectors of collided individuals are exchanged. The equation of the threshold is defined as follows:

$$\kappa = \frac{\sum_{j=1}^{n}(b_j^{high} - b_j^{low})}{n} * \lambda, \tag{8}$$

where κ is the threshold and λ is a constant positive between 0 and 1. The collision operation formula is defined as follows:

$$temp = d_j, d_j = d_i, d_i = temp; \tag{9}$$

Random behaviour: Random behaviour is another measure to maintain the diversity of the population, and it can effectively prevent the population from falling into local optima. Compared with the other two operations, random behaviour is not necessary. Each individual may or may not perform a random behaviour. Random behaviour is expressed as follows:

$$P_{i,j}^{m+1} = \begin{cases} b_j^{low} + R_2 * (b_j^{high} - b_j^{low}), & \text{with probability } H; \\ P_{i,j}^{m+1}, & \text{with probability (1-}H\text{),} \end{cases} \tag{10}$$

where R_2 is a random number similar to R_1 and H denotes the occurrence probability of random behaviour.

The whole solution search process in the SMS algorithm needs to go through three major phases: the gas phase, liquid phase and solid phase. The parameters η, λ, ρ and H are set to different values in the different phases. In this study, these parameters are set to the defaults according to [35].

2.5.2. Scale-free local search and the BA algorithm

The scale-free local search has a complex topological structure. Fig. 3 shows a schematic diagram of a scale-free network. In such a scale-free network, there are fewer nodes with a high vertex degree than nodes with a low vertex degree, and the distribution follows a power law, which can be expressed by the following equation [40]:

$$P(l) \propto l^{-\gamma}, \tag{11}$$

where $p(l)$ denotes the probability of a node possessing degree l and γ represents a scaling exponent that is commonly in the range of [2,3]. Notably, $p(l)$ and $l^{-\gamma}$ are positively correlated. To intuitively observe such a law, the curve of its distribution is plotted in Fig. 4. As shown in Fig. 4, when the degree increases, the number of nodes decreases. In addition, the distribution is presented as a straight line in the coordinate axes of the logarithmic scale. Previous studies have shown that it is difficult to map complex networks in the real world with only a scale-free network [41]. To solve this problem, the Barabasi–Albert (BA) algorithm, proposed by Barabasi and Albert [40], is adopted to construct the scale-free network. The BA algorithm is inspired by the links formed between the new nodes and the original nodes in a real-world network, which can better reflect the real environment. In the BA algorithm, the newly generated nodes are connected to the old nodes with a particular preference.

The detailed process of the BA algorithm is as follows: (1) Initialize the scale-free network with a fully connected network with M_0 nodes. Set the total number of nodes N. (2) Generate a novel node j and calculate the probability P_i for all existing nodes with the equation $P_i = \delta(i)/\sum_j \delta(j)$, where $\delta(i)$ represents the degree of node i. (3) Generate a new link between novel node j and existing node i with probability P_i. (4) Repeat steps (1)–(3) until all nodes are connected to the network. Moreover, by proving that $P(l)$ is proportional to l^{-3}, Barabasi and Albert identified that the scale-free network in the BA algorithm still abides by the power-law distribution. In such a distribution space, links are more likely to appear between low-degree nodes and high-degree nodes. γ is also utilized to denote the extent to which low-degree nodes link to high-degree nodes. The equation for γ can be expressed as follows [42]:

$$\gamma = \frac{N^{-1}\sum_n p_n q_n - (N^{-1}\sum_n (p_n + q_n)/2)^2}{N^{-1}\sum_n (p_n^2 + q_n^2)/2 - (N^{-1}\sum_n (p_n + q_n)/2)^2} \tag{12}$$

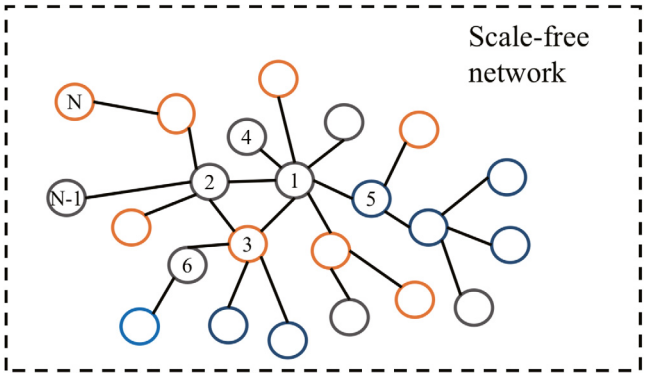

Fig. 3. Schematic diagram of the scale-free network.

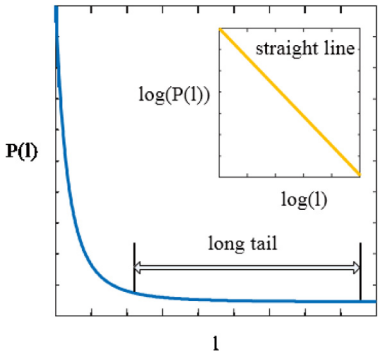

Fig. 4. The degree distribution curve of a scale-free network.

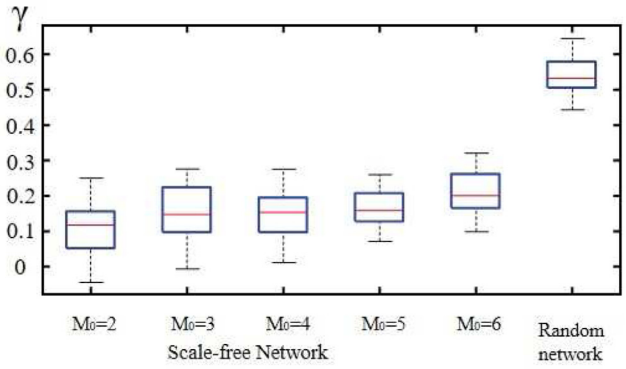

Fig. 5. Degree–degree correlation coefficients γ of different networks.

where N denotes the number of links in the scale-free network architecture. p_n and q_n are the degrees of the two adjacent nodes at both ends of the nth link. When the value of γ is high, links between nodes with a high degree are more likely to be generated. In addition, we also investigate the effect of the initial node number M_0 on γ. The experimental result is shown in Fig. 5. It should be emphasized that we conducted 30 experiments for all networks. As shown in Fig. 5, the values of γ are lower in the scale-free networks than in the random network, and the value of γ increases as M_0 increases. When M_0 is 2, the value of γ achieves the lowest value.

2.5.3. Scale-free local search

In this study, we use a scale-free local search as a new component of the SMS algorithm (termed SFSMS) to further enhance the solution searching capability. First, a corresponding scale-free network is generated using the BA algorithm in terms of the size of the population. Second, we number each node in the network. According to the rules of the BA algorithm, there are more low-degree nodes and fewer high-degree nodes in the network. Third, in each iteration, we rank the individuals in terms of their fitness in the population after the SMS operations. Then, we put each individual into a network node with the same number and rank. As a result, high-degree nodes store excellent individuals, and low-degree nodes store poor individuals. Finally, we update each individual utilizing the following equation:

$$Y_{j+1}^{i} = Y_j^i + rand(0,1)(Y_j^{neighbour} - Y_j^i), \tag{13}$$

where Y_j^i represents the weight vector of the ith individual in the jth generation after the SMS operation. $Y_j^{neighbour}$ denotes an arbitrary node in the scale-free network that is linked to the node of Y_j^i. There are two main advantages in using the scale-free network. On the one hand, high-quality individuals are always stored in the high-degree nodes, and the power-law distribution reduces the number of high-degree nodes. Thus, most individuals are close to excellent individuals after updating, which improves the level of the whole population and accelerates the convergence of the model. On the other hand, since the value of γ is relatively low in the BA algorithm, the probability of a link between high-degree nodes is low; namely, excellent individuals are less likely to attract each other, which ensures the diversity of the population and prevents falling into locally optimal situations. The detailed content of the SFSMS algorithm is introduced in Algorithm 1, and the flowchart for the overall methodology is demonstrated in Fig. 6.

3. Study of the chaotic time series

Before applying DNR to forecast the transmission trend of the COVID-19 pandemic, the raw data should be preprocessed. According to Takens's theorem [36], the hidden information of the time series can be revealed by the time delay and the embedding dimensions. Utilizing these two components, we can reconstruct the phase space of the data, which can greatly enhance the precision of forecasting. In addition, it is necessary to use the maximum Lyapunov exponent (λ_{max}) to determine that the dataset is a chaotic time series when we use this method. The reconstructed dataset is available only when λ_{max} exceeds zero.

3.1. Phase space reconstruction technique

In this study, the PSR technique proposed by Takens is adopted for data preprocessing. It is very difficult to predict the trend of a one-dimensional chaotic time series because of its randomness and intermittency. The characteristics of a one-dimensional chaotic time series are distinctive because there is some high-dimensional information that can hardly be described in one dimension. Even with ML methods, it is difficult to effectively describe high-dimensional information. Through PSR, we reconstruct the representations of the one-dimensional data so that their hidden laws can be revealed. Then, through ML methods to fit the representations, we can achieve a satisfactory effect. According to Takens's theorem, the reconstruction operation requires two key parameters, i.e., the time delay σ and embedding dimension ω. Suppose $\{y(i)|i = 1, 2, \ldots, N\}$ is the raw one-dimensional chaotic time series. In terms of the two parameters σ and ω, the reconstructed data can be expressed as follows:

$$Y_i = (y_i, y_{i+\sigma}, y_{i+2\sigma}, \ldots, y_{i+(\omega-1)\sigma})^T, \tag{14}$$

$$T_i = y_{i+(\omega-1)\sigma-1}, \tag{15}$$

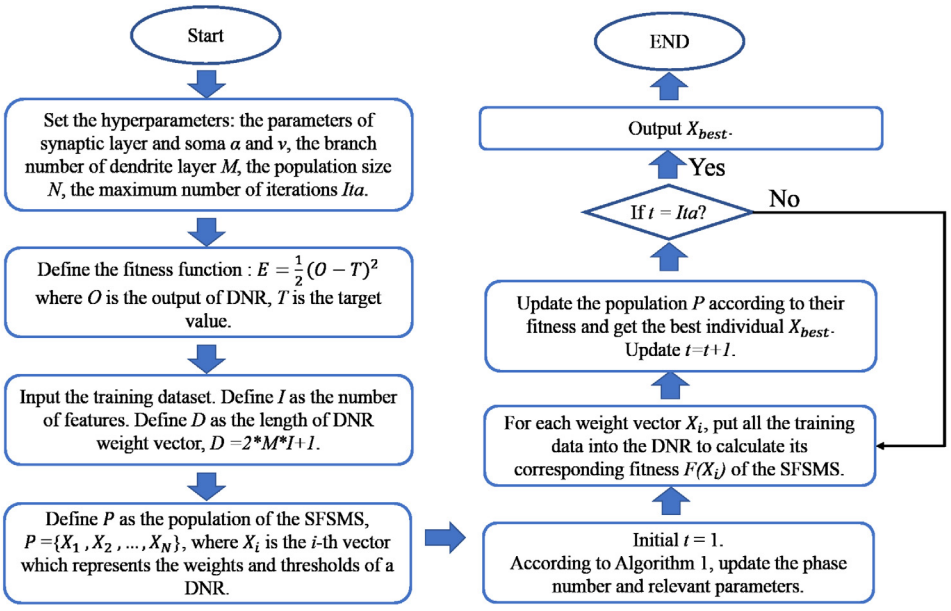

Fig. 6. The flowchart of the DNR-SFSMS.

where Y_i is the ith input vector and T_i is the ith target for the neural model. When the two parameters σ and ω are set reasonably, Y_i can clearly describe the laws of the chaotic time series. The process for creating Y_i refers to the PSR technique.

Obviously, the key of the PSR technique is to choose suitable approaches to obtain the time delay and embedding dimensions. However, since the time series always carry different kinds of noise, there is no method that can effectively calculate σ and ω for every time series. In most cases, we have to choose the appropriate method based on experience. After a certain trial, the mutual information (MI) algorithm [43] and the false nearest neighbour (FNN) algorithm [44] are employed to obtain the time delay and embedding dimensions, respectively.

3.2. Mutual information algorithm

In this study, the time delay is obtained using the MI algorithm, which is one of the most effective algorithms for such a problem. Let $H(X)$, which is called the information entropy, be the degree of uncertainty of X. The function of $H(X)$ is given as follows:

$$H(X) = -\sum_{i=1}^{n} P(x_i) log P(x_i), \tag{16}$$

where X is a condition set, x_i denotes the ith condition, $P(x_i)$ represents the probability of condition x_i, and n is the length of X, i.e., the number of conditions. Let $H(X|Y)$ be the conditional information entropy, which can be expressed as follows:

$$H(X|Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(y_j)P(x_i|y_j) log P(x_i|y_j), \tag{17}$$

where Y is another condition set, m denotes the length of Y, $P(y_j)$ is the probability of the jth condition in Y, and $P(x_i|y_j)$ is the probability of the ith condition in X occurring under the jth condition in Y. Suppose $I(X, Y)$ is the mutual information entropy and is calculated as follows:

$$I(X, Y) = H(X) + H(Y) - H(X, Y), \tag{18}$$

where $H(X, Y)$ is called the joint formation of X and Y and is defined as follows:

$$H(X, Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) log P(x_i, y_j). \tag{19}$$

According to the rule of the MI algorithm, let $\{z(i)|i = 1, 2, \ldots, N\}$ represent the time series. Therefore, the mutual information entropy $I(z_p, z_{p+\sigma})$ can be given as follows:

$$I(\sigma) = I(z_p, z_{p+\sigma}) = H(z_p) + H(z_{p+\sigma}) - H(z_p, z_{p+\sigma}). \tag{20}$$

The components of this function were introduced previously. When the value of $I(\sigma)$ reaches a local minimum for the first time, the value of σ is the required time delay.

3.3. False nearest neighbours algorithm

The embedding dimension is an important component of PSR. If the embedding dimension is not selected properly, the hidden information in the high-dimensional space can hardly be exhibited in the one-dimensional space, and the laws in the time series cannot be extracted effectively. In this study, the FNN algorithm proposed by Kennel [43] is employed to obtain the embedding dimension for the COVID-19 pandemic data. In terms of the theory of the FNN algorithm, false nearest neighbour points are defined as two adjacent points that are a large distance apart in high-dimensional space. It is difficult to utilize these two points to demonstrate the hidden laws in the data. Thus, we gradually increase the embedding dimensions so that the trajectory in the high-dimensional space is clearer, which can help in obtaining a highly accurate prediction. When the embedding dimension reaches a certain value, the trajectory in the higher dimension is fully exhibited; this is the value that we seek. Let ω be the number of embedding dimensions and $h_i(\omega) = (x(i), x(\sigma + i), \ldots, x((\omega - 1)\sigma + i))$ be an ω-dimensional vector in the phase space. Suppose $h_j(\omega)$ is the nearest neighbour point of h_i. Then, the distance between $h_i(\omega)$ and $h_j(\omega)$ can be expressed as follows:

$$Dis_i(\omega) = \|h_i(\omega) - h_j(\omega)\|. \tag{21}$$

It is obvious that the value of $Dis_i(\omega)$ varies with increasing ω. When ω is increased by one, the updated $Dis_i(\omega+1)$ can be given

Algorithm 1: Pseudocode for the SFSMS algorithm.

Input: Population size N, number of dimensions n, maximum number of iterations Ita, initial number of nodes M_0.
Result: Best individual $\bar{P}$.
begin
  Initialize the population $\mathcal{X} = \{P_1, \ldots, P_N\}$, the direction vector set $\mathcal{D}^0 = \{d_1^0, \ldots, d_N^0\}$, the maximum number of iterations $Ita = 1000$, the state count $phase = 1$, and the current iteration number $m = 1$;
  repeat
    if $phase == 1$ **then**
      $\eta = 0.8$, $\rho = 0.8$, $\lambda = 0.9$, $H = 0.9$, $PD = Ita * 0.5$;
    if $phase == 2$ **then**
      $\eta = 0.4$, $\rho = 0.2$, $\lambda = 0.5$, $H = 0.2$, $PD = Ita * 0.9$;
    if $phase == 3$ **then**
      $\eta = 0.1$, $\rho = 0.0$, $\lambda = 0.0$, $H = 0.0$, $PD = Ita$;
    for $m, m <= PD, m = m + 1$ **do**
      Calculate the fitness function of the population $\mathcal{F}(\mathcal{X}) = \{f(P_1), \ldots, f(P_N)\}$;
      Set the current best individual to $\bar{P}$;
      /* Perform direction vector operations */
      Calculate the new direction vector set $\mathcal{D}^t$ with Eq. (5);
      Utilize $\mathcal{D}^t$ to obtain the velocity vector set $\mathcal{V} = \{v_1, \ldots, v_N\}$ with Eq. (6);
      Update the population with Eq. (7);
      /* Perform the collision operations */
      Calculate the threshold κ in terms of Eq. (8);
      Calculate the distance between each pair of individuals; if the distance is less than r, exchange the direction vectors of the individuals by Eq. (9);
      /* Perform random behaviour */
      Obtain a random number $\bar{r}$ in the range of [0,1]; if H exceeds $\bar{r}$, random behaviour is executed with Eq. (10);
      /* Utilize the scale-free network component*/
      Generate a scale-free network with the BA algorithm and number each node;
      Calculate the fitness function of the population $\mathcal{F}(\mathcal{X}) = \{f(P_1), \ldots, f(P_N)\}$;
      Rank each individual by fitness;
      Put each individual into a network node with the same number and rank;
      Update each individual by Eq. (13);
    phase=phase+1;
  until $phase > 3$;
return The best individual $\bar{P}$.

as follows:

$$Dis_i^2(\omega + 1) = Dis_i^2(\omega) + \|h_i(i + \omega * \sigma) - h_j(j + \omega * \sigma)\|. \quad (22)$$

These two points are regarded as false nearest neighbour points when $Dis_i(\omega)$ is much less than $Dis_i(\omega + 1)$. This function can be expressed in another form as follows:

$$Dis_\sigma = \frac{\|h_i(i + \omega * \sigma) - h_j(j + \omega * \sigma)\|}{Dis_i(\omega)}. \quad (23)$$

Let θ be a threshold in the range of [10,50]. The two points can be considered false nearest neighbour points when θ is less than Dis_σ. In this study, the initial number of embedding dimensions ω is set to 2. Then, ω is increased until the proportion of false nearest neighbour points is less than 5%, which is the value of ω that we require. It should be emphasized that in some cases, the proportion of false nearest neighbour points cannot reach 5%. To solve this problem, we set an upper bound of ω. In this study, the upper bound of ω is set to 8.

3.4. Maximum Lyapunov exponent

The maximum Lyapunov exponent (λ_{max}) is utilized to identify the chaotic characteristics of the historical COVID-19 data. Only when the chaotic characteristics are confirmed can the reconstructed data be used for the ML methods. According to [45], when λ_{max} of the reconstructed data exceeds zero, the data can be seen as chaotic. In this study, Wolf's method [46], one of the most effective methods for calculating λ_{max}, is employed to obtain the value.

Suppose the vector $Y(t) = (y(t), y(\sigma + t), \ldots, y((\omega - 1)\sigma + t))$ is a reconstructed series through the PSR technique. Let $Y(t_i)$ and $Y(t_j)$ be the two closest points, and their distance can be expressed as follows:

$$R_i = \|Y(t_j) - Y(t_i)\|. \quad (24)$$

Then, we can use the component to calculate λ_{max}, and the equation is defined as follows:

$$\lambda_{max} = \frac{1}{t_M - t_0} \sum_{i=0}^{M} ln \frac{R_i}{R_i'}, M = N - (\omega - 1)\sigma, \quad (25)$$

where t_0 is the initial time and t_M is the final time. In addition, the recommended length for the time series prediction can be expressed as follows [47,48]:

$$\Delta t = \frac{1}{\lambda_{max}}. \quad (26)$$

4. Experiment and discussion

This section is mainly divided into the following contents: First, the relevant data and environment of the experiment are presented. Second, we introduce in detail the steps of the normalization operation and the special treatment of this operation in this problem. Third, the evaluation indicators and charts are described. Finally, the experimental results are presented and discussed.

4.1. Benchmark datasets

In this study, DNR, a novel data-driven method trained by the SFSMS algorithm (DNR-SFSMS), is utilized to forecast the transmission trend of the COVID-19 pandemic. The whole process is shown in Fig. 7. The variation in the number of confirmed COVID-19 cases in six countries from March 1, 2020, to March 12, 2021, is collected as experimental data for analysing DNR performance. The six countries include India, Angola, Indonesia, Ethiopia, Azerbaijan and Israel. These data are available for free at https://datahub.io, a website called Datahub that is dedicated to discovering and sharing high-quality data sets, which was accessed in March 14, 2021. The data were collected by Johns Hopkins University through various public approaches, and the collated data were uploaded to GitHub. The variation in the number of confirmed COVID-19 cases in six countries is shown in Fig. 8.

These datasets were first processed by the PSR technique and converted into a group of new feature vector sets and target sets. It should be emphasized that the lengths of the feature vector sets and target sets may differ for each country because the selected σ and ω may differ in the process of reconstruction. The reconstructed data from each country were split into two portions. The first portion contained 300 pieces of data utilized as the training data, and the rest of the data were employed as the test data. Then, the model was trained via the normalized training data, and its performance was evaluated via the normalized test data. All experiments were conducted on a personal PC with an Intel(R) Core *i*7, 2.90 GHz, and 16 GB memory using MATLAB R2020a.

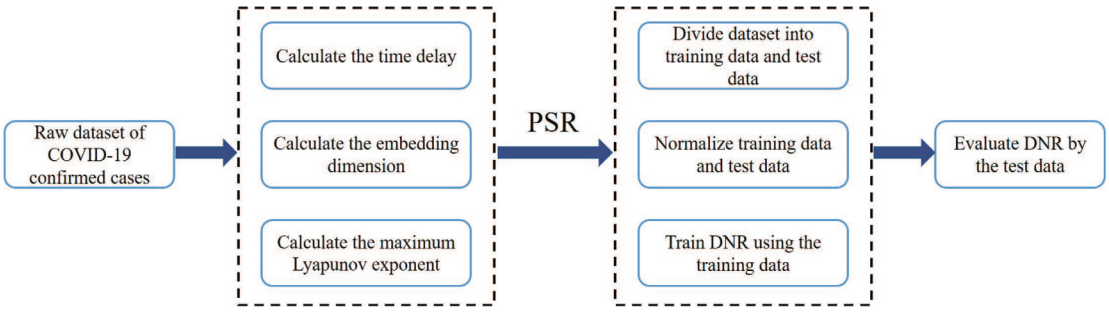

Fig. 7. The flowchart of the transmission trend of COVID-19 pandemic prediction.

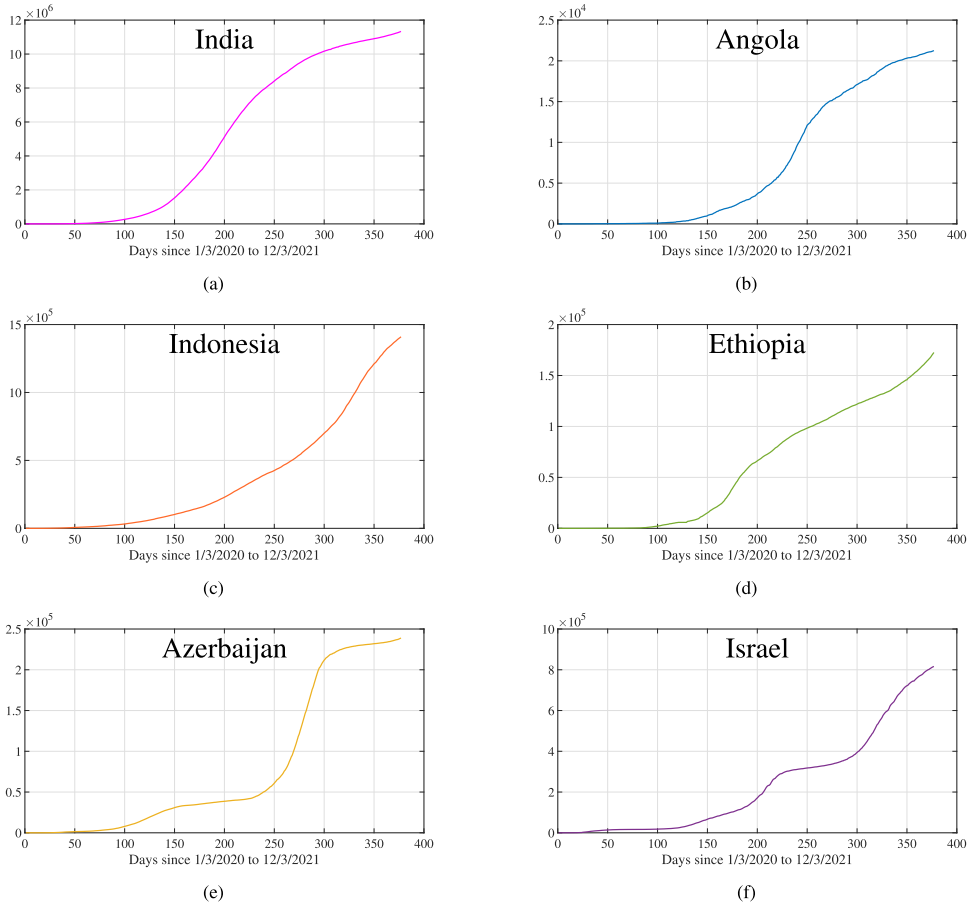

Fig. 8. The growth curves of the numbers of COVID-19 cases diagnosed in each country.

Table 1
Results for the time delay, embedding dimensionality and maximum Lyapunov exponents of the time series confirmed COVID-19 case data from six countries.

Country	Time delay (τ)	Embedding dimensionality (m)	Maximum Lyapunov exponent (λ_{max})	Chaotic	Training data size	Test data size
India	1	2	0.0182	Yes	300	75
Angola	1	6	0.0332	Yes	300	71
Indonesia	1	4	0.0198	Yes	300	73
Ethiopia	1	3	0.0252	Yes	300	74
Azerbaijan	5	4	0.0152	Yes	300	61
Israel	3	2	0.0226	Yes	300	73

4.2. Normalization

The purpose of the normalization operation is to reduce the computational time and increase the accuracy of prediction. Normalization is performed not only on the training data but also on the test data. There are several approaches to perform a normalization operation, such as mean and variance normalization and simple normalization [49,50]. In this study, the normalization function can be expressed as follows:

$$\bar{y} = \frac{y - MIN(y)}{MAX(y) - MIN(y)}(b - a) + a, \tag{27}$$

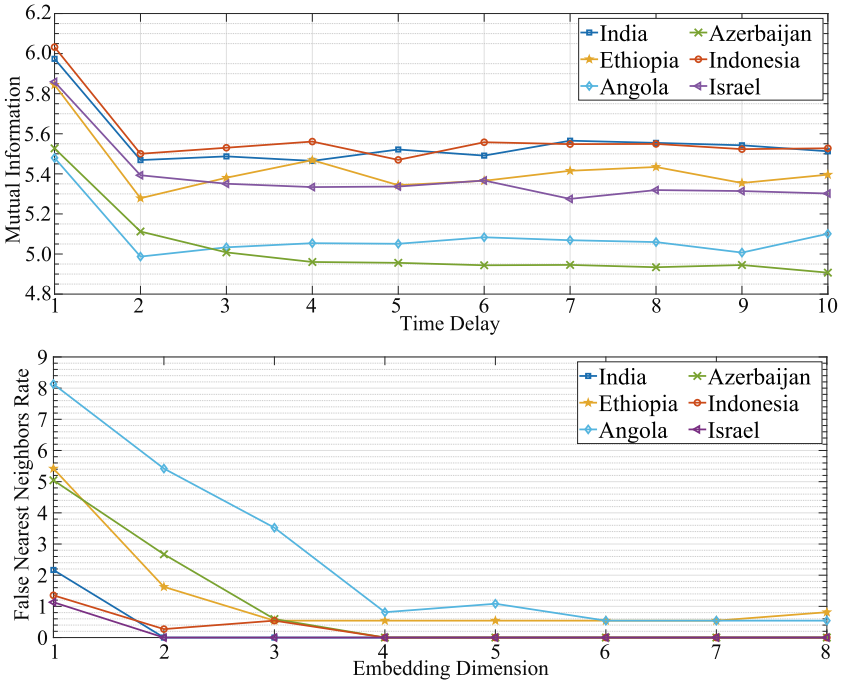

Fig. 9. The results of the MI algorithm and FNN algorithm on the time series confirmed COVID-19 case data from six countries.

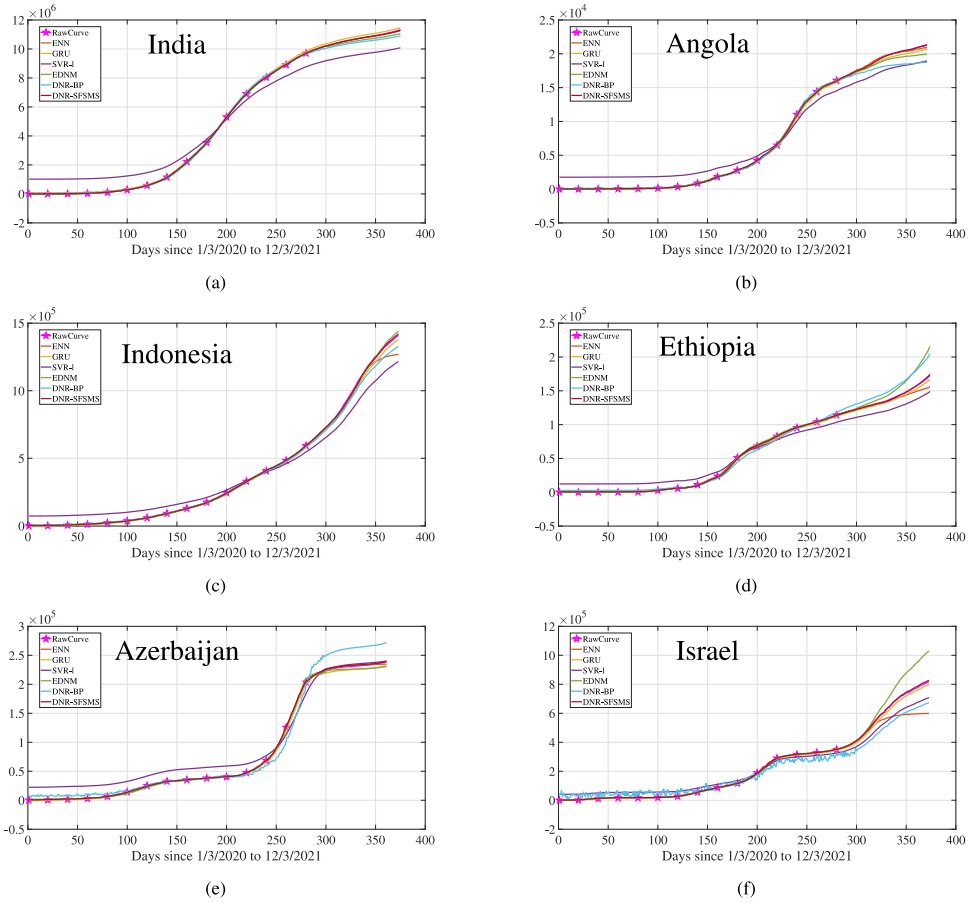

Fig. 10. Training and prediction results of DNR-SFSMS for six countries.

where y is a member of the reconstructed feature vectors or a target value. $\bar{y}$ is the normalized value of y. $MAX(y)$ and $MIN(y)$ are the maximum value and the minimum value of y, respectively. b and a are the upper bound and the lower bound, respectively. Usually, b defaults to 1, and a defaults to 0. Since DNR employs a sigmoid function as the activation function in the last layer, the

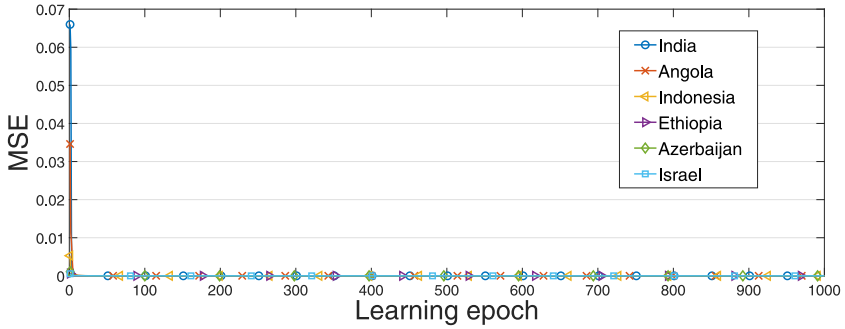

Fig. 11. Convergence curves of DNR-SFSMS for transmission trend prediction of the COVID-19 pandemic.

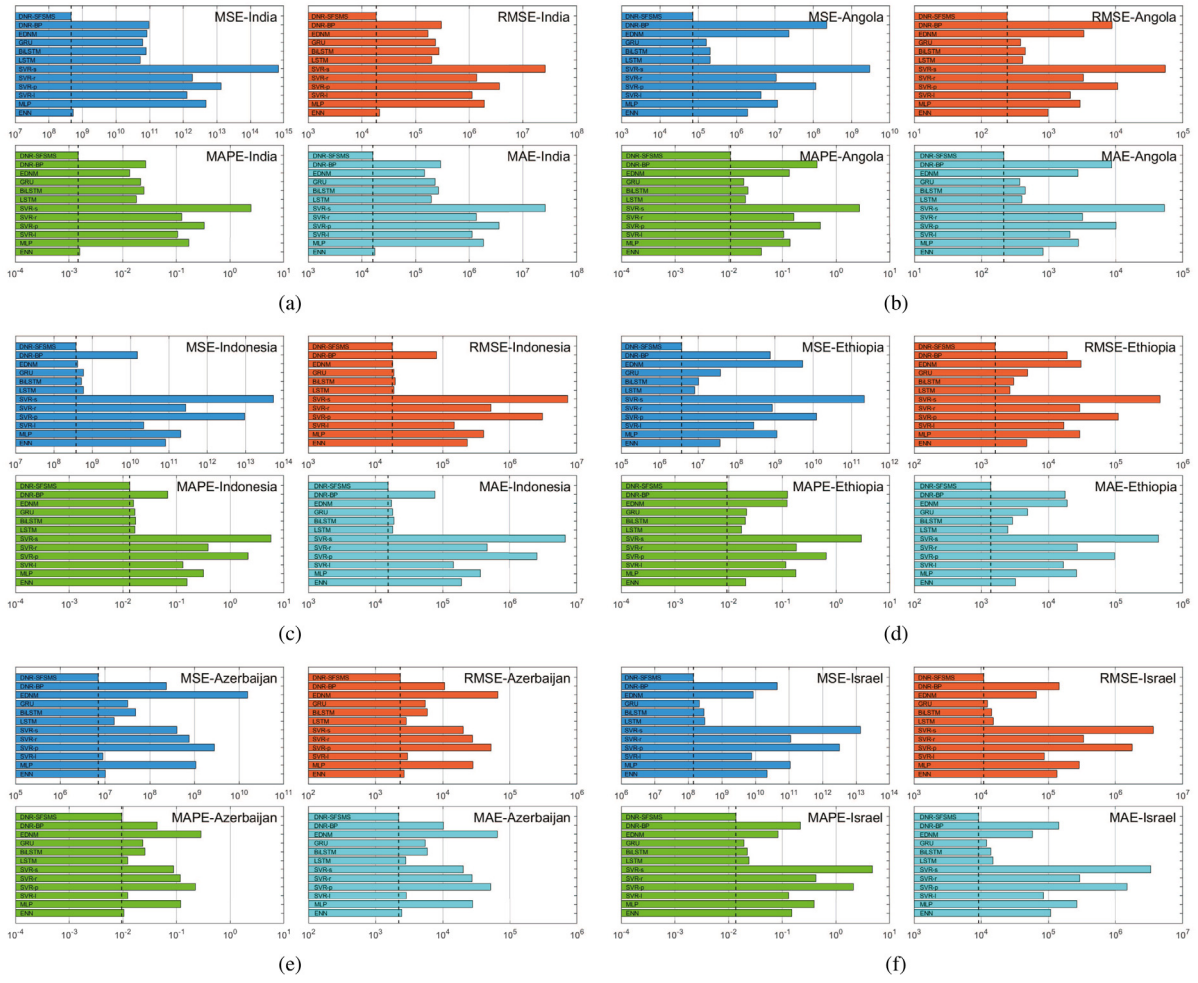

Fig. 12. Comparison of the four error metrics of prediction methods on the data from the six countries. From the top, the methods are DNR-SFSMS, DNR-BP, EDNM, GRU, BiLSTM, LSTM, SVR-s, SVR-r, SVR-p, SVR-l, MLP and ENN, respectively.

Table 2
Results for the time delay, embedding dimensionality, maximum Lyapunov exponents and the specific allocation of training data and test data of the dataset of the COVID-19 pandemic in different countries.

Country	*k*	*M*	*qs*	*epoch*	Normalization range	popsize (EDNM,DNR-SFSMS)	Learning rate (DNR-BP)
India	5	6	0.5	1000	[0.3,0.65]	100	0.03
Angola	6	3	0.5	1000	[0.2,0.5]	100	0.05
Indonesia	5	5	0.5	1000	[0.4,0.469]	100	0.01
Ethiopia	6	7	0.5	1000	[0.1,0.19]	100	0.01
Azerbaijan	6	4	0.5	1000	[0.15,0.255]	100	0.05
Israel	5	4	0.5	1000	[0.3,0.313]	100	0.12

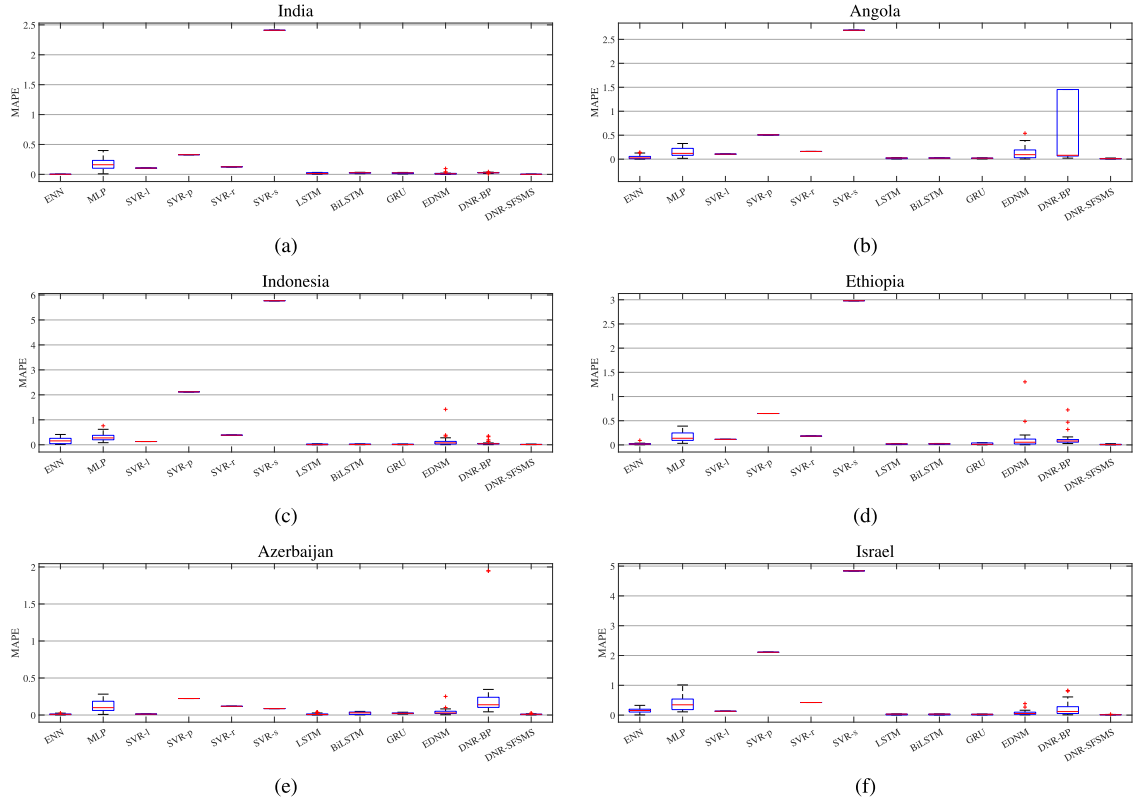

Fig. 13. Box-and-whisker plots for the MAPE of DNR-SFSMS for six countries.

Table 3
Parameter settings of the algorithms for the transmission trend prediction of the COVID-19 pandemic.

Algorithm	Parameters
MLP	*HiddenLayer* = 10, *learningRate* = 0.01, *epoch* = 1000
ENN	*learningRate* = 0.01, *epoch* = 1000
SVR	linear, polynomial, RBF and sigmoid kernels
LSTM, BiLSTM, GRU	*HiddenUnits* = 200, *epoch* = 1000

output of DNR is in the range of [0,1]. Thus, the normalization operation is an imperative step when applying DNR. In addition, for the monotonically increasing problem of predicting the transmission trend of the COVID-19 pandemic, the selected values of the upper bound and lower bound strongly influence the results.

4.3. Evaluation metrics

To fairly compare the performance differences between DNR-SFSMS and other algorithms, several evaluation approaches were utilized in this study:

(1) Evaluation metrics: In this study, four commonly used evaluation metrics were adopted for our experiments: the mean square error (*MSE*), the root mean square error (*RMSE*), the mean absolute percentage error (*MAPE*), and the mean absolute error (*MAE*). Their equations can be expressed as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(L - \bar{L})^2, \tag{28}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(L - \bar{L})^2}, \tag{29}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\bar{L} - L}{\bar{L}}\right|, \tag{30}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\bar{L} - L|, \tag{31}$$

where $\bar{L}$ represents the desired values and L denotes the prediction outputs. *MSE*, *RMSE*, *MAPE* and *MAE* are all error-based metrics; thus, lower values of these metrics indicate better performance of the algorithm.

(2) Relative charts: The charts plotted during the experiments can help to intuitively observe the performance of each algorithm and more easily draw a conclusion for the result. Several commonly used charts were generated to compare the performance of all algorithms. First, fitting charts were drawn to show the difference between the desired curve and the prediction curve. Second, a convergence chart was plotted to demonstrate the convergence effect and speed of DNR-SFSMS on all datasets. Third, histograms were generated to reveal the metric differences. Finally, boxplots were produced to display the entire situation in the experiments.

(3) Nonparametric statistical test: The purpose of the test is to determine whether there is a significant difference between DNR-SFSMS and other ML methods. The Wilcoxon rank-sum test [51, 52] was adopted in this study and implemented in terms of the KEEL software [53]. The significance level was set to 5%, which indicates that there is a significant difference in performance between the DNR-SFSMS and another algorithm when the *p-value* is less than 5%.

4.4. Experimental results and discussion

In terms of the methods mentioned above, the three components σ, ω and λ_{max} of PSR for the time series of confirmed

Table 4
Prediction performance for confirmed COVID-19 cases in India, Angola and Indonesia.

India								
Model	*MSE* (*Mean* ± *Std*)	*p-value*	*RMSE* (*Mean* ± *Std*)	*p-value*	*MAPE* (*Mean* ± *Std*)	*p-value*	*MAE* (*Mean* ± *Std*)	*p-value*
ENN	5.33E+08 ± 3.80E+08	1.33E−01	2.12E+04 ± 9.16E+03	1.21E−01	1.61E−03 ± 7.23E−04	2.05E−01	1.76E+04 ± 7.95E+03	1.83E−01
MLP	4.78E+12 ± 4.98E+12	9.13E−07	1.89E+06 ± 1.09E+06	9.13E−07	1.71E−01 ± 1.01E−01	9.13E−07	1.85E+06 ± 1.09E+06	9.13E−07
SVR-l	1.28E+12 ± 0.00E+00	9.13E−07	1.13E+06 ± 0.00E+00	9.13E−07	1.05E−01 ± 0.00E+00	9.13E−07	1.13E+06 ± 0.00E+00	9.13E−07
SVR-p	1.34E+13 ± 0.00E+00	9.13E−07	3.65E+06 ± 0.00E+00	9.13E−07	3.29E−01 ± 0.00E+00	9.13E−07	3.57E+06 ± 0.00E+00	9.13E−07
SVR-r	1.88E+12 ± 0.00E+00	9.13E−07	1.37E+06 ± 0.00E+00	9.13E−07	1.26E−01 ± 0.00E+00	9.13E−07	1.36E+06 ± 0.00E+00	9.13E−07
SVR-s	6.90E+14 ± 0.00E+00	9.13E−07	2.63E+07 ± 0.00E+00	9.13E−07	2.41E+00 ± 0.00E+00	9.13E−07	2.60E+07 ± 0.00E+00	9.13E−07
LSTM	5.16E+10 ± 4.73E+10	9.13E−07	1.99E+05 ± 1.09E+05	9.13E−07	1.80E−02 ± 1.01E−02	9.13E−07	1.95E+05 ± 1.11E+05	9.13E−07
BiLSTM	7.73E+10 ± 3.27E+10	9.13E−07	2.71E+05 ± 6.41E+04	9.13E−07	2.48E−02 ± 5.87E−03	9.13E−07	2.68E+05 ± 6.42E+04	9.13E−07
GRU	6.23E+10 ± 3.81E+10	9.13E−07	2.35E+05 ± 8.50E+04	9.13E−07	2.13E−02 ± 7.85E−03	9.13E−07	2.31E+05 ± 8.58E+04	9.13E−07
EDNM	8.24E+10 ± 2.69E+11	1.85E−06	1.70E+05 ± 2.31E+05	1.67E−06	1.35E−02 ± 1.82E−02	3.02E−06	1.48E+05 ± 1.99E+05	3.02E−06
DNR-BP	9.64E+10 ± 4.56E+10	9.13E−07	3.01E+05 ± 7.55E+04	9.13E−07	2.68E−02 ± 6.79E−03	9.13E−07	2.91E+05 ± 7.36E+04	9.13E−07
DNR-SFSMS	**4.62E+08 ± 5.20E+08**	–	**1.85E+04 ± 1.09E+04**	–	**1.48E−03 ± 9.54E−04**	–	**1.60E+04 ± 1.04E+04**	–
Angola								
Model	*MSE* (*Mean* ± *Std*)	*p-value*	*RMSE* (*Mean* ± *Std*)	*p-value*	*MAPE* (*Mean* ± *Std*)	*p-value*	*MAE* (*Mean* ± *Std*)	*p-value*
ENN	1.92E+06 ± 3.30E+06	1.07E−04	9.85E+02 ± 9.77E+02	1.47E−04	4.04E−02 ± 3.99E−02	2.75E−04	8.24E+02 ± 8.15E+02	2.54E−04
MLP	1.17E+07 ± 1.27E+07	9.13E−07	2.93E+03 ± 1.78E+03	9.13E−07	1.38E−01 ± 8.65E−02	9.13E−07	2.76E+03 ± 1.73E+03	9.13E−07
SVR-l	4.36E+06 ± 0.00E+00	9.13E−07	2.09E+03 ± 0.00E+00	9.13E−07	1.05E−01 ± 0.00E+00	9.13E−07	2.08E+03 ± 0.00E+00	9.13E−07
SVR-p	1.17E+08 ± 0.00E+00	9.13E−07	1.08E+04 ± 0.00E+00	9.13E−07	5.07E−01 ± 0.00E+00	9.13E−07	1.02E+04 ± 0.00E+00	9.13E−07
SVR-r	1.09E+07 ± 0.00E+00	9.13E−07	3.30E+03 ± 0.00E+00	9.13E−07	1.61E−01 ± 0.00E+00	9.13E−07	3.21E+03 ± 0.00E+00	9.13E−07
SVR-s	3.03E+09 ± 0.00E+00	9.13E−07	5.50E+04 ± 0.00E+00	9.13E−07	2.69E+00 ± 0.00E+00	9.13E−07	5.36E+04 ± 0.00E+00	9.13E−07
LSTM	2.06E+05 ± 1.44E+05	4.99E−04	4.13E+02 ± 1.89E+02	4.99E−04	2.02E−02 ± 9.59E−03	2.54E−04	4.01E+02 ± 1.93E+02	3.45E−04
BiLSTM	2.06E+05 ± 5.06E+04	2.04E−06	4.50E+02 ± 5.61E+01	1.51E−06	2.26E−02 ± 2.81E−03	9.13E−07	4.46E+02 ± 5.60E+01	1.01E−06
GRU	1.64E+05 ± 9.53E+04	5.37E−04	3.82E+02 ± 1.33E+02	7.16E−04	1.89E−02 ± 6.72E−03	1.47E−04	3.73E+02 ± 1.35E+02	1.87E−04
EDNM	2.27E+07 ± 4.14E+07	4.03E−06	3.32E+03 ± 3.41E+03	3.66E−06	1.33E−01 ± 1.29E−01	3.02E−06	2.72E+03 ± 2.66E+03	3.02E−06
DNR-BP	2.22E+08 ± 3.63E+08	9.13E−07	8.78E+03 ± 1.20E+04	9.13E−07	4.40E−01 ± 6.11E−01	9.13E−07	8.71E+03 ± 1.20E+04	9.13E−07
DNR-SFSMS	**7.17E+04 ± 6.97E+04**	–	**2.42E+02 ± 1.16E+02**	–	**1.07E−02 ± 5.26E−03**	–	**2.14E+02 ± 1.07E+02**	–
Indonesia								
Model	*MSE* (*Mean* ± *Std*)	*p-value*	*RMSE* (*Mean* ± *Std*)	*p-value*	*MAPE* (*Mean* ± *Std*)	*p-value*	*MAE* (*Mean* ± *Std*)	*p-value*
ENN	8.10E+10 ± 9.01E+10	9.13E−07	2.32E+05 ± 1.64E+05	9.13E−07	1.56E−01 ± 1.11E−01	9.13E−07	1.91E+05 ± 1.36E+05	9.13E−07
MLP	2.04E+11 ± 1.90E+11	9.13E−07	4.08E+05 ± 1.92E+05	9.13E−07	3.13E−01 ± 1.54E−01	9.13E−07	3.64E+05 ± 1.79E+05	9.13E−07
SVR-l	2.22E+10 ± 0.00E+00	9.13E−07	1.49E+05 ± 0.00E+00	9.13E−07	1.30E−01 ± 0.00E+00	9.13E−07	1.45E+05 ± 0.00E+00	9.13E−07
SVR-p	9.54E+12 ± 0.00E+00	9.13E−07	3.09E+06 ± 0.00E+00	9.13E−07	2.12E+00 ± 0.00E+00	9.13E−07	2.57E+06 ± 0.00E+00	9.13E−07
SVR-r	2.77E+11 ± 0.00E+00	9.13E−07	5.26E+05 ± 0.00E+00	9.13E−07	3.85E−01 ± 0.00E+00	9.13E−07	4.57E+05 ± 0.00E+00	9.13E−07
SVR-s	5.43E+13 ± 0.00E+00	9.13E−07	7.37E+06 ± 0.00E+00	9.13E−07	5.77E+00 ± 0.00E+00	9.13E−07	6.70E+06 ± 0.00E+00	9.13E−07
LSTM	5.96E+08 ± 6.34E+08	1.42E−01	1.90E+04 ± 1.53E+04	3.33E−01	1.68E−02 ± 1.38E−02	1.01E−01	1.82E+04 ± 1.53E+04	1.62E−01
BiLSTM	5.17E+08 ± 5.06E+08	9.13E−07	1.98E+04 ± 1.11E+04	9.13E−07	1.74E−02 ± 1.06E−02	9.13E−07	1.89E+04 ± 1.14E+04	9.13E−07
GRU	4.20E+08 ± 3.63E+08	3.18E−01	1.81E+04 ± 9.54E+03	4.19E−01	1.59E−02 ± 9.24E−03	1.01E−01	1.73E+04 ± 9.88E+03	1.77E−01
EDNM	2.54E+11 ± 1.08E+12	1.85E−06	2.43E+05 ± 4.42E+05	1.67E−06	1.43E−01 ± 2.55E−01	1.67E−06	1.80E+05 ± 3.25E+05	1.67E−06
DNR-BP	1.49E+10 ± 3.61E+10	2.04E−06	8.07E+04 ± 9.13E+04	3.02E−06	6.87E−02 ± 8.16E−02	2.25E−06	7.69E+04 ± 8.97E+04	2.48E−06
DNR-SFSMS	**3.78E+08 ± 3.34E+08**	–	**1.78E+04 ± 7.81E+03**	–	**1.33E−02 ± 5.68E−03**	–	**1.54E+04 ± 6.76E+03**	–

COVID-19 cases in six countries were calculated. The results of PSR operation and the specific allocation of training data and test data are displayed in Table 1. The maximum Lyapunov exponents of the six countries exceed zero, which indicates that these time series are chaotic. Thus, the reconstructed datasets can be used in the prediction. In addition, the results of the time delays and the embedding dimensions calculations are plotted in Fig. 9.

For a comprehensive evaluation, several commonly used ML methods are adopted for comparison with the DNR-SFSMS. These methods include the multilayer perceptron (MLP) [54]; the Elman neural network (ENN) [55]; support vector regression with a linear kernel (SVR-l), a polynomial kernel (SVR-p), an RBF kernel (SVR-r) and a sigmoid kernel (SVR-s) [56]; RNN variants, including LSTM, BiLSTM and GRU; the DNM trained by L-SHADE (EDNM) [57]; and the original DNM. In addition, the random search method [58] is utilized to adjust the hyperparameters of the algorithms in this study. The range of normalization is also regarded as one of the hyperparameters of the DNM-based methods since it severely influences the prediction accuracy. The hyperparameter settings of the DNM-based methods and other algorithms are shown in Tables 2 and 3, respectively. All the experiments for each algorithm were independently conducted 30 times for each dataset.

Fig. 10 shows the fitting charts for the COVID-19 trend forecasts in six countries. For the four SVR-based algorithms, SVR-l, which has the best performance, is selected as the representative. For the three RNN-based algorithms, GRU is selected as the representative since it has a better result than the other two. It should be emphasized that the reconstructed time series of the number of confirmed COVID-19 cases in each country were divided into a training dataset comprising the first 300 pieces of data and a test dataset comprising the remaining data. As shown in Fig. 10, in the training dataset, namely, the first 300 days, all algorithms except SVR-l have a very good fitting effect. However, for the test dataset, the difference between the curve generated by DNR-SFSMS and the original curve is significantly smaller than those of the other algorithms. Fig. 11 shows the convergence effect of DNR-SFSMS when trained on the six datasets. As shown in Fig. 11, although the maximum iteration number is set to 1000, DNR-SFSMS always converges within 100 generations, and the convergence effects are outstanding. The results indicate that the SFSMS algorithm has excellent performance in optimizing the weight of DNR and that DNR has good generalization for this type of problem.

The results of the four performance metrics on the six datasets for each algorithm in the form of "average ± standard deviation" are reported in Tables 4 and 5. For each metric, we present the best results in bold. As shown in Tables 4 and 5, DNR-SFSMS achieves the best results for any dataset, which implies that DNR-SFSMS has greater effectiveness and stability in regard to COVID-19 trend prediction. Moreover, comparing DNR-SFSMS, EDNM and the original DNM, we can find that, on the one hand, adding a weight on the dendrite layer to describe the strength of the dendrite branches is indeed conducive to the regression ability. In addition, compared with the traditional BP and L-SHADE algorithms to optimize the model, the SFSMS algorithm has better adaptability with the DNM-based models, the SFSMS

Table 5
Prediction performance for confirmed COVID-19 cases in Ethiopia, Azerbaijan and Israel .

Ethiopia								
Model	*MSE* (*Mean* ± *Std*)	*p-value*	*RMSE* (*Mean* ± *Std*)	*p-value*	*MAPE* (*Mean* ± *Std*)	*p-value*	*MAE* (*Mean* ± *Std*)	*p-value*
ENN	3.70E+07 ± 8.43E+07	2.77E−05	4.74E+03 ± 3.81E+03	3.03E−05	2.06E−02 ± 1.60E−02	3.71E−04	3.22E+03 ± 2.52E+03	2.75E−04
MLP	1.14E+09 ± 1.09E+09	9.13E−07	2.93E+04 ± 1.67E+04	9.13E−07	1.78E−01 ± 1.11E−01	9.13E−07	2.62E+04 ± 1.61E+04	9.13E−07
SVR-l	2.84E+08 ± 0.00E+00	9.13E−07	1.69E+04 ± 0.00E+00	9.13E−07	1.15E−01 ± 0.00E+00	9.13E−07	1.66E+04 ± 0.00E+00	9.13E−07
SVR-p	1.21E+10 ± 0.00E+00	9.13E−07	1.10E+05 ± 0.00E+00	9.13E−07	6.49E−01 ± 0.00E+00	9.13E−07	9.66E+04 ± 0.00E+00	9.13E−07
SVR-r	8.58E+08 ± 0.00E+00	9.13E−07	2.93E+04 ± 0.00E+00	9.13E−07	1.82E−01 ± 0.00E+00	9.13E−07	2.68E+04 ± 0.00E+00	9.13E−07
SVR-s	2.16E+11 ± 0.00E+00	9.13E−07	4.65E+05 ± 0.00E+00	9.13E−07	2.98E+00 ± 0.00E+00	9.13E−07	4.36E+05 ± 0.00E+00	9.13E−07
LSTM	8.21E+06 ± 5.07E+06	2.75E−03	2.68E+03 ± 1.01E+03	1.25E−03	1.73E−02 ± 6.44E−03	2.54E−04	2.49E+03 ± 9.88E+02	7.69E−04
BiLSTM	1.02E+07 ± 5.35E+06	6.00E−05	3.06E+03 ± 8.94E+02	2.54E−05	2.05E−02 ± 5.66E−03	6.49E−06	2.93E+03 ± 8.73E+02	1.03E−05
GRU	3.82E+07 ± 4.39E+07	7.16E−04	4.92E+03 ± 3.74E+03	4.31E−04	2.13E−02 ± 1.63E−02	6.41E−03	4.89E+03 ± 3.75E+03	9.08E−05
EDNM	5.33E+09 ± 2.44E+10	2.74E−06	3.04E+04 ± 6.64E+04	2.25E−06	1.22E−01 ± 2.39E−01	3.02E−06	1.91E+04 ± 3.82E+04	3.02E−06
DNR-BP	7.59E+08 ± 2.08E+09	9.13E−07	1.89E+04 ± 2.00E+04	9.13E−07	1.23E−01 ± 1.41E−01	9.13E−07	1.78E+04 ± 2.01E+04	9.13E−07
DNR-SFSMS	**3.72E+06 ± 4.55E+06**	–	**1.62E+03 ± 1.04E+03**	–	**9.28E−03 ± 6.31E−03**	–	**1.38E+03 ± 9.39E+02**	–
Azerbaijan								
Model	*MSE* (*Mean* ± *Std*)	*p-value*	*RMSE* (*Mean* ± *Std*)	*p-value*	*MAPE* (*Mean* ± *Std*)	*p-value*	*MAE* (*Mean* ± *Std*)	*p-value*
ENN	1.01E+07 ± 1.25E+07	1.12E−01	2.67E+03 ± 1.71E+03	1.38E−01	1.05E−02 ± 6.84E−03	1.94E−01	2.44E+03 ± 1.59E+03	1.83E−01
MLP	1.08E+09 ± 1.12E+09	1.01E−06	2.80E+04 ± 1.73E+04	1.01E−06	1.19E−01 ± 7.52E−02	1.01E−06	2.76E+04 ± 1.74E+04	1.01E−06
SVR-l	8.86E+06 ± 0.00E+00	6.79E−03	2.98E+03 ± 0.00E+00	5.38E−03	1.23E−02 ± 0.00E+00	4.50E−03	2.85E+03 ± 0.00E+00	5.70E−03
SVR-p	2.75E+09 ± 0.00E+00	9.12E−07	5.24E+04 ± 0.00E+00	9.12E−07	2.23E−01 ± 0.00E+00	9.12E−07	5.17E+04 ± 0.00E+00	9.12E−07
SVR-r	7.57E+08 ± 0.00E+00	9.12E−07	2.75E+04 ± 0.00E+00	9.12E−07	1.18E−01 ± 0.00E+00	9.12E−07	2.74E+04 ± 0.00E+00	9.12E−07
SVR-s	4.08E+08 ± 0.00E+00	9.12E−07	2.02E+04 ± 0.00E+00	9.12E−07	8.69E−02 ± 0.00E+00	9.12E−07	2.01E+04 ± 0.00E+00	9.12E−07
LSTM	1.61E+07 ± 2.52E+07	5.97E−01	2.87E+03 ± 2.81E+03	6.13E−01	1.22E−02 ± 1.22E−02	5.41E−01	2.83E+03 ± 2.83E+03	5.41E−01
BiLSTM	4.89E+07 ± 4.46E+07	1.87E−04	5.92E+03 ± 3.73E+03	2.02E−04	2.55E−02 ± 1.62E−02	1.73E−04	5.91E+03 ± 3.74E+03	1.87E−04
GRU	3.21E+07 ± 1.62E+07	1.01E−06	5.49E+03 ± 1.39E+03	1.01E−06	2.36E−02 ± 6.02E−03	1.01E−06	5.48E+03 ± 1.39E+03	1.01E−06
EDNM	2.37E+08 ± 6.90E+08	8.59E−06	1.06E+04 ± 1.12E+04	7.82E−06	4.37E−02 ± 4.53E−02	1.03E−05	1.02E+04 ± 1.05E+04	1.03E−05
DNR-BP	1.53E+10 ± 5.03E+10	9.13E−07	6.54E+04 ± 1.05E+05	9.13E−07	2.82E−01 ± 4.53E−01	9.13E−07	6.53E+04 ± 1.05E+05	9.13E−07
DNR-SFSMS	**7.08E+06 ± 7.79E+06**	–	**2.33E+03 ± 1.29E+03**	–	**9.49E−03 ± 5.60E−03**	–	**2.21E+03 ± 1.30E+03**	–
Israel								
Model	*MSE* (*Mean* ± *Std*)	*p-value*	*RMSE* (*Mean* ± *Std*)	*p-value*	*MAPE* (*Mean* ± *Std*)	*p-value*	*MAE* (*Mean* ± *Std*)	*p-value*
ENN	2.20E+10 ± 1.80E+10	1.01E−06	1.35E+05 ± 6.22E+04	1.01E−06	1.51E−01 ± 7.31E−02	1.12E−06	1.09E+05 ± 5.26E+04	1.12E−06
MLP	1.07E+11 ± 1.07E+11	9.13E−07	2.89E+05 ± 1.53E+05	9.13E−07	3.90E−01 ± 2.24E−01	9.13E−07	2.65E+05 ± 1.50E+05	9.13E−07
SVR-l	7.61E+09 ± 0.00E+00	9.13E−07	8.72E+04 ± 0.00E+00	9.13E−07	1.30E−01 ± 0.00E+00	9.13E−07	8.51E+04 ± 0.00E+00	9.13E−07
SVR-p	3.18E+12 ± 0.00E+00	9.13E−07	1.78E+06 ± 0.00E+00	9.13E−07	2.11E+00 ± 0.00E+00	9.13E−07	1.50E+06 ± 0.00E+00	9.13E−07
SVR-r	1.11E+11 ± 0.00E+00	9.13E−07	3.33E+05 ± 0.00E+00	9.13E−07	4.23E−01 ± 0.00E+00	9.13E−07	2.95E+05 ± 0.00E+00	9.13E−07
SVR-s	1.36E+13 ± 0.00E+00	9.13E−07	3.69E+06 ± 0.00E+00	9.13E−07	4.84E+00 ± 0.00E+00	9.13E−07	3.34E+06 ± 0.00E+00	9.13E−07
LSTM	3.11E+08 ± 2.39E+08	3.31E−03	1.53E+04 ± 8.71E+03	1.54E−02	2.38E−02 ± 1.41E−02	1.17E−03	1.49E+04 ± 9.02E+03	2.92E−03
BiLSTM	2.87E+08 ± 2.50E+08	1.46E−02	1.43E+04 ± 9.12E+03	4.79E−02	2.21E−02 ± 1.50E−02	6.41E−03	1.39E+04 ± 9.34E+03	1.01E−02
GRU	2.09E+08 ± 1.78E+08	5.66E−02	1.25E+04 ± 7.34E+03	2.00E−01	1.94E−02 ± 1.22E−02	1.46E−02	1.21E+04 ± 7.53E+03	2.79E−02
EDNM	8.59E+09 ± 1.75E+10	1.85E−06	6.62E+04 ± 6.48E+04	1.85E−06	8.34E−02 ± 8.14E−02	1.51E−06	5.79E+04 ± 5.70E+04	1.67E−06
DNR-BP	4.43E+10 ± 8.07E+10	1.01E−06	1.45E+05 ± 1.52E+05	1.01E−06	2.17E−01 ± 2.28E−01	1.12E−06	1.42E+05 ± 1.49E+05	1.01E−06
DNR-SFSMS	**1.42E+08 ± 1.27E+08**	–	**1.09E+04 ± 4.78E+03**	–	**1.36E−02 ± 5.13E−03**	–	**9.13E+03 ± 3.80E+03**	–

algorithm can search the global optimal solution more effectively, and its searching ability is more stable. Moreover, most of the *p-value*s in the tables are less than 5%. According to the rules of the nonparametric statistics test, we can determine that the precision of DNR-SFSMS is significantly better than those of the other algorithms.

In Fig. 12, the performance metrics are presented in the form of bar charts. On each chart, we use a black dotted line to highlight the performance differences between DNR-SFSMS and the other algorithms. From Fig. 12, we can intuitively observe that DNR-SFSMS yields the best results on all datasets. Through further observation, we find that the predictions are more accurate for countries with more infections, that is, countries with more severe outbreaks such as India, than for countries with less severe outbreaks. The situation is mainly reflected in the fact that the MAPE values are smaller. We speculate that this is because, for countries with mild outbreaks, the country has effectively suppressed the transmission of the disease through some policies, so it does not conform to the law of natural transmission of the disease, resulting in a decline in the prediction accuracy of the model. Since MAPE is more of a reference than the other performance metrics for such large numerical prediction problems, we employ box plots to show the results of MAPE in Fig. 13. The box plots can help us visualize the detailed circumstance of each dataset and each algorithm in 30 runs. In the box plots, the ordinate represents the MAPE value, and the rectangle represents the range of the overall MAPE values of the algorithms in the experiment. A rectangle with a large area denotes great fluctuations and poor stability of the algorithm. In addition, the red plus sign indicates the algorithm fell into a local optimum. The more signs there are, the more times the experiment falls into a local optimum. It should be noted that due to the nature of SVR-based algorithms, the results of each run of all SVR-based algorithms are the same, so they are represented as a line in box plots. Fig. 13 shows that DNR-SFSMS has almost the minimum rectangular area on all datasets, the positions of the rectangles on the coordinate axis are the lowest, and it has the least number of red plus signs. These situations suggest that the forecasting precision of DNR in COVID-19 disease trend prediction is superior to the forecasting precision of other commonly used ML methods. In conclusion, the above experimental results suggest that DNR-SFSMS can be used as a competitive tool for COVID-19 trend prediction.

5. Conclusion

Since the outbreak of the novel COVID-19 pandemic, machine learning techniques have played a vital role in pandemic prevention, contact tracing, rapid screening and the development of vaccines and drugs. In this study, a novel DNR approach is applied to forecast the transmission trend of COVID-19 pandemics. The regression capability of DNR is enhanced by employing the combination of the SMS algorithm and scale-free local search. The SMS algorithm is a recently proposed evolutionary algorithm with strong optimization ability, and the scale-free local

search can improve the quality of the population during evolution. Since the COVID-19 pandemic dataset can be deemed a time series, Takens's theorem is applied to the data to improve the prediction accuracy. We utilize the time delay calculated by the MI algorithm and the embedding dimensions calculated via the FNN algorithm to implement the PSR operation. Then, the maximum Lyapunov exponent is calculated to determine the chaos of the reconstructed data, which affirms the availability of the reconstructed data. To fairly evaluate the proposed method, several commonly used ML methods are employed as competitors. The experimental results show that DNR-SFSMS is more competitive than the other methods for forecasting the COVID-19 transmission trend in terms of various evaluation metrics. DNR-SFSMS can be regarded as a powerful prediction approach. Our future research will focus on applying DNR-SFSMS to forecast the transmission of COVID-19 in more countries and simultaneously validate its forecast performance on other infectious diseases.

CRediT authorship contribution statement

Minhui Dong: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Cheng Tang:** Resources, Validation, Investigation, Writing - review & editing. **Junkai Ji:** Methodology, Validation, Formal analysis, Investigation, Writing - review & editing, Supervision. **Qiuzhen Lin:** Resources, Investigation, Writing - review & editing. **Ka-Chun Wong:** Resources, Supervision, Project administration.

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.

Acknowledgements

This work was supported by the National Key R&D Program of China under Grant 2020YFA0908700, the Guangdong "Pearl River Talent Recruitment Program", China under Grant 2019ZT08X603, Shenzhen Science and Technology Innovation Commission, China (R2020A045), a Project of the Guangdong Basic and Applied Basic Research Fund, China (No. 2019A1515111139), the National Natural Science Foundation of China under Grant 61876110, the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212, and the Shenzhen Technology Plan, China under Grant JCYJ20190808164211203.

References

[1] S. Ying, F. Li, X. Geng, Z. Li, X. Du, H. Chen, S. Chen, M. Zhang, Z. Shao, Y. Wu, et al., Spread and control of COVID-19 in China and their associations with population movement, public health emergency measures, and medical resources, 2020, MedRxiv.
[2] Y. Bai, L. Yao, T. Wei, F. Tian, D.-Y. Jin, L. Chen, M. Wang, Presumed asymptomatic carrier transmission of COVID-19, JAMA 323 (14) (2020) 1406–1407.
[3] 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. (2020).
[4] D. Houssin, T. Ghebreyesus, K.J. Yang, J. Lanche, K. Kupferschmidt, WHO emergencies coronavirus emergency committee second meeting, 30 january 2020, in: C. Lindmeier (Ed.), Coronavirus Disease (COVID-2019) Press Briefings, World Health Organization, Geneva, Switzerland, 2020.
[5] N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han, Y. Qiu, J. Wang, Y. Liu, Y. Wei, et al., Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in wuhan, China: a descriptive study, Lancet 395 (10223) (2020) 507–513.
[6] W.H. Organization, et al., COVID-19 Weekly Epidemiological Update, World Health Organization, 2020.
[7] E. Guidotti, D. Ardia, COVID-19 data hub, J. Open Source Softw. 5 (51) (2020) 2376.
[8] D. Nokes, R. Anderson, The use of mathematical models in the epidemiological study of infectious diseases and in the design of mass immunization programmes, Epidemiol. Infect. 101 (1) (1988) 1–20.
[9] O. Sharomi, C.N. Podder, A.B. Gumel, B. Song, Mathematical analysis of the transmission dynamics of HIV/TB coinfection in the presence of treatment, Math. Biosci. Eng. 5 (1) (2008) 145.
[10] A. Basing, S. Tay, Malaria transmission dynamics of the Anopheles mosquito in Kumasi, Ghana, Int. J. Infect. Dis. 21 (2014) 22.
[11] N. Chitnis, J.M. Cushing, J. Hyman, Bifurcation analysis of a mathematical model for malaria transmission, SIAM J. Appl. Math. 67 (1) (2006) 24–45.
[12] A. Jajarmi, A. Yusuf, D. Baleanu, M. Inc, A new fractional HRSV model and its optimal control: a non-singular operator approach, Physica A 547 (2020) 123860.
[13] D. Baleanu, A. Jajarmi, H. Mohammadi, S. Rezapour, A new study on the mathematical modelling of human liver with Caputo–Fabrizio fractional derivative, Chaos Solitons Fractals 134 (2020) 109705.
[14] S. Zhao, Q. Lin, J. Ran, S.S. Musa, G. Yang, W. Wang, Y. Lou, D. Gao, L. Yang, D. He, et al., Preliminary estimation of the basic reproduction number of novel coronavirus (2019-nCoV) in China, from 2019 to 2020: A data-driven analysis in the early phase of the outbreak, Int. J. Infect. Dis. 92 (2020) 214–217.
[15] E. Shim, A. Tariq, W. Choi, Y. Lee, G. Chowell, Transmission potential and severity of COVID-19 in South Korea, Int. J. Infect. Dis. (2020).
[16] D. Benvenuto, M. Giovanetti, L. Vassallo, S. Angeletti, M. Ciccozzi, Application of the ARIMA model on the COVID-2019 epidemic dataset, Data in Brief (2020) 105340.
[17] T. Dehesh, H. Mardani-Fard, P. Dehesh, Forecasting of covid-19 confirmed cases in different countries with arima models, 2020, MedRxiv.
[18] D. Baleanu, A. Jajarmi, S.S. Sajjadi, J.H. Asad, The fractional features of a harmonic oscillator with position-dependent mass, Commun. Theor. Phys. 72 (5) (2020) 055002.
[19] T.A. Yıldız, A. Jajarmi, B. Yıldız, D. Baleanu, New aspects of time fractional optimal control problems within operators with nonsingular kernel, Discrete Contin. Dyn. Syst. S 13 (3) (2020) 407.
[20] A. Jajarmi, D. Baleanu, S.S. Sajjadi, J.H. Asad, A new feature of the fractional Euler–Lagrange equations for a coupled oscillator using a nonsingular operator approach, Front. Phys. 7 (2019) 196.
[21] S. Kombrink, T. Mikolov, M. Karafiát, L. Burget, Recurrent neural network based language modeling in meeting recognition, in: Twelfth Annual Conference of the International Speech Communication Association, 2011.
[22] J. Schmidhuber, S. Hochreiter, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[23] 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.
[24] V.K.R. Chimmula, L. Zhang, Time series forecasting of COVID-19 transmission in Canada using LSTM networks, Chaos Solitons Fractals (2020) 109864.
[25] S. Shastri, K. Singh, S. Kumar, P. Kour, V. Mansotra, Time series forecasting of Covid-19 using deep learning models: India-USA comparative case study, Chaos Solitons Fractals 140 (2020) 110227.
[26] S.K. Bandyopadhyay, S. Dutta, Machine learning approach for confirmation of covid-19 cases: Positive, negative, death and release, 2020, MedRxiv.
[27] C.-J. Huang, Y.-H. Chen, Y. Ma, P.-H. Kuo, Multiple-input deep convolutional neural network model for covid-19 forecasting in China, 2020, MedRxiv.
[28] A. Tomar, N. Gupta, Prediction for the spread of COVID-19 in India and effectiveness of preventive measures, Sci. Total Environ. (2020) 138762.
[29] R. Pal, A.A. Sekh, S. Kar, D.K. Prasad, Neural network based country wise risk prediction of COVID-19, 2020, arXiv preprint arXiv:2004.00959.
[30] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2016) 2222–2232.
[31] J. Ji, S. Song, Y. Tang, S. Gao, Z. Tang, Y. Todo, Approximate logic neuron model trained by states of matter search algorithm, Knowl.-Based Syst. 163 (2019) 120–130.
[32] T. Zhou, S. Gao, J. Wang, C. Chu, Y. Todo, Z. Tang, Financial time series prediction using a dendritic neuron model, Knowl.-Based Syst. 105 (2016) 214–224.
[33] W. Chen, J. Sun, S. Gao, J.-J. Cheng, J. Wang, Y. Todo, Using a single dendritic neuron to forecast tourist arrivals to Japan, IEICE Trans. Inf. Syst. 100 (1) (2017) 190–202.
[34] J.K. Makara, A. Losonczy, Q. Wen, J.C. Magee, Experience-dependent compartmentalized dendritic plasticity in rat hippocampal CA1 pyramidal neurons, Nature Neurosci. 12 (12) (2009) 1485.
[35] E. Cuevas, A. Echavarría, M.A. Ramírez-Ortegón, An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation, Appl. Intell. 40 (2) (2014) 256–272.
[36] F. Takens, Detecting strange attractors in turbulence, in: Dynamical Systems and Turbulence, Warwick 1980, Springer, 1981, pp. 366–381.

[37] E. Salinas, L.F. Abbott, A model of multiplicative neural responses in parietal cortex, Proc. Natl. Acad. Sci. 93 (21) (1996) 11956–11961.
[38] F. Gabbiani, H.G. Krapp, C. Koch, G. Laurent, Multiplicative computation in a visual neuron sensitive to looming, Nature 420 (6913) (2002) 320–324.
[39] J. Ji, S. Gao, J. Cheng, Z. Tang, Y. Todo, An approximate logic neuron model with a dendritic structure, Neurocomputing 173 (2016) 1775–1783.
[40] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[41] A.D. Broido, A. Clauset, Scale-free networks are rare, 2018, arXiv preprint arXiv:1801.03400.
[42] M.E. Newman, Assortative mixing in networks, Phys. Rev. Lett. 89 (20) (2002) 208701.
[43] A.M. Fraser, H.L. Swinney, Independent coordinates for strange attractors from mutual information, Phys. Rev. A 33 (2) (1986) 1134.
[44] M.B. Kennel, R. Brown, H.D. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction, Phys. Rev. A 45 (6) (1992) 3403.
[45] C. Grebogi, E. Ott, J.A. Yorke, Crises, sudden changes in chaotic attractors, and transient chaos, Physica D 7 (1–3) (1983) 181–200.
[46] A. Wolf, J.B. Swift, H.L. Swinney, J.A. Vastano, Determining Lyapunov exponents from a time series, Physica D 16 (3) (1985) 285–317.
[47] P. Shang, X. Na, S. Kamae, Chaotic analysis of time series in the sediment transport phenomenon, Chaos Solitons Fractals 41 (1) (2009) 368–379.
[48] H. Abarbanel, Analysis of Observed Chaotic Data, Springer Science & Business Media, 2012.
[49] J. Sola, J. Sevilla, Importance of input data normalization for the application of neural networks to complex industrial problems, IEEE Trans. Nucl. Sci. 44 (3) (1997) 1464–1468.
[50] G. Zhang, B.E. Patuwo, M.Y. Hu, Forecasting with artificial neural networks:: The state of the art, Int. J. Forecast. 14 (1) (1998) 35–62.
[51] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, J. Heuristics 15 (6) (2009) 617.
[52] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[53] J. Alcalá-Fdez, L. Sanchez, S. Garcia, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, et al., KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (3) (2009) 307–318.
[54] M. London, M. Häusser, Dendritic computation, Annu. Rev. Neurosci. 28 (2005) 503–532.
[55] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[56] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 1–27.
[57] Z. Song, Y. Tang, J. Ji, Y. Todo, Evaluating a dendritic neuron model for wind speed forecasting, Knowl.-Based Syst. (2020) 106052.
[58] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (1) (2012) 281–305.