Neural Computing and Applications (2023) 35:15397–15413
https://doi.org/10.1007/s00521-023-08513-0

ORIGINAL ARTICLE

Prediction of PM2.5 time series by seasonal trend decomposition-based dendritic neuron model

Zijing Yuan[1] · Shangce Gao[1] · Yirui Wang[2] · Jiayi Li[1] · Chunzhi Hou[1] · Lijun Guo[2]

Received: 19 October 2022 / Accepted: 21 March 2023 / Published online: 11 April 2023

Abstract

The rapid industrial development in the human society has brought about the air pollution, which seriously affects human health. PM2.5 concentration is one of the main factors causing the air pollution. To accurately predict PM2.5 microns, we propose a dendritic neuron model (DNM) trained by an improved state-of-matter heuristic algorithm (DSMS) based on STL-LOESS, namely DS-DNM. Firstly, DS-DNM adopts STL-LOESS for the data preprocessing to obtain three characteristic quantities from original data: seasonal, trend, and residual components. Then, DNM trained by DSMS predicts the residual values. Finally, three sets of feature quantities are summed to obtain the predicted values. In the performance test experiments, five real-world PM2.5 concentration data are used to test DS-DNM. On the other hand, four training algorithms and seven prediction models were selected for comparison to verify the rationality of the training algorithms and the accuracy of the prediction models, respectively. The experimental results show that DS-DNM has the more competitive performance in PM2.5 concentration prediction problem.

Keywords Dendritic neuron model · PM2.5 concentration prediction · Evolutionary algorithms · Preprocessing technology

1 Introduction

Among various pollutants, high concentrations of fine particulate matter (PM2.5 and fine aerosols with particle size less than or equal to 2.5 microns) seriously affect the human health. Previous studies show that the increase of PM2.5 concentration can reduce the lung function in humans, damage the respiratory system, and increase the incidence of cardiovascular disease [1], and it can also bring about environmental problems such as haze [2].

Monitoring and reducing the air pollution has become a development strategy for many countries. Many regions have established observatories to monitor the PM2.5 concentration in real time. By monitoring the pollution concentration over a period of time and predicting future concentrations, governments can make timely and appropriate countermeasures to protect the health and safety of the public. Thus, accurate and timely prediction of PM2.5 concentration is crucial. Since there are many factors affecting PM2.5 concentration and its variation is irregular and nonlinear, it is very difficult to predict the concentration accurately. Currently, air quality prediction methods can be classified into three types: deterministic methods, statistical methods, and machine learning methods [3].

Deterministic methods have fully functional relationships based on known physical and chemical theories about the contaminant transfer. For example, the community multiscale air quality (CMAQ) modeling system [4] based on the atmospheric physical and chemical processes uses

✉ Shangce Gao
gaosc@eng.u-toyama.ac.jp

Zijing Yuan
tzyuanzijing@gmail.com

Yirui Wang
wangyirui@nbu.edu.cn

Jiayi Li
lijiayi212@gmail.com

Chunzhi Hou
hcz604791531@gmail.com

Lijun Guo
guolijun@nbu.edu.cn

[1] Faculty of Engineering, University of Toyama, Toyama-shi 930-8555, Japan

[2] Engineering and Computer Science, Ningbo University, Zhejiang 315221, China

meteorological principles and mathematical methods to simulate the air quality and reproduce the transport, reabsorption, and removal processes of pollutants in the atmosphere for prediction. Similar to CMAQ, many air quality prediction models based on deterministic methods have been proposed, such as the CHIMERE model designed for seasonal simulation and prediction [5], and the LOTOS model [6] proposed by Dutch scholars. However, there are many factors affecting PM2.5 concentration, such as carbon, nitrogen, and sulfur contents. Their relationships are complex [7]. Deterministic models often require a large number of computational resources [8].

Statistical methods generally use regression and time series methods for the prediction. A PM2.5 prediction model using the land use regression method has been proposed. It is developed for seasonal, pedestrian flow, and other factors and gets good results for the air quality prediction in Beijing of China [9]. A statistical model based on linear multiple regression and a categorical regression tree analysis method is used for the air quality prediction in Macau of China. Experimental results show that the column method is reasonable for the prediction with high concentrations and low confidence levels [10]. However, due to the complexity of air quality models, many influencing factors, and the irregular and nonlinear variation of data, traditional statistical methods often fail to obtain accurate results. To address this problem, researchers have proposed various machine learning methods.

Nowadays, machine learning methods such as multilayer perceptron (MLP), support vector machine (SVM), and long short-term memory (LSTM) have obtained good results in the field of air quality prediction [11–14]. A decomposition-ensemble broad learning system, which innovatively proposes a dynamic decomposition time window based on three decomposition methods, combined with a broad learning system, achieves excellent performance for air quality index forecasting [15]. Considering the unique changes in population migration in the post-COVID-19 pandemic age and the correlation between population migration and air quality index, a migration attentive graph convolutional network accurately predicted PM2.5 concentrations in Hubei Province by combining migration data [16]. A novel predictive neural network (LSTM-FC) using a temporal simulator based on LSTM captures the PM2.5 concentration relationship between neighboring spatial stations through the spatial combination based on neural networks. The better experimental results are obtained for the PM2.5 concentration prediction problem in Beijing of China [17]. Three different methods, i.e., multilayer neural networks, linear regression, and persistence, are used to predict PM2.5 concentrations in downtown Santiago of Chile. This study selects months with high values of concentrations for the experiment. Results show that the neural network model has the more competitive power [18]. In six regions of China, a WT-SAE-LSTM model is proposed. It shows that some progress is got in addressing the problem of easy gradient disappearance of LSTM [19]. Applying deep learning to predict spatio-temporal sequence problems has become a hot topic, but the huge model size makes it difficult for practical applications. At the same time, deep learning models rely heavily on a large amount of training data, and results are often unsatisfactory in the face of data-scarce fields.

In this paper, an artificial neural network called dendritic neuron model (DNM) is used to predict the PM2.5 concentration. The DNM model has a simple structure with four layers. The dendrite layer and membrane layer make the model have the ability of automatic pruning. Since its proposal, DNM has achieved excellent results in various problems such as photovoltaic power generation forecasting, financial time series forecasting, and tourism number forecasting [20, 21]. However, the original DNM model is trained by BP [22]. The gradient descent-based BP algorithm easily falls into local optima; thus, the appropriate training algorithm is crucial to improve the prediction ability of DNM model. Evolutionary algorithms are good choices and have been used in many fields [23, 24]. The original SMS algorithm is selected to train the DNM model, which improves the prediction accuracy of the model [25]. We propose an algorithm based on an innovative population structure selection method (DSMS) for the DNM model. DSMS deals with the population structure again based on the state-of-matter heuristic algorithm (SMS). After sorting, classifying and cross-variation operations, according to different selection strategies, the excellent individuals are retained and the poorer ones are updated, which better balances the exploration and exploitation of the algorithm.

In addition, due to the chaotic and nonlinear characteristics of PM2.5 time series, a preprocessing technique STL-LOESS is used [26]. The seasonal trend decomposition

Table 1 Parameters of DS-DNM

State	State weight (%)	β	λ	η	Random probability H
Gaseous	50	0.8	0.8	[0.8, 1.0]	0.9
Liquid	40	0.4	0.2	[0.0, 0.6]	0.2
Solid	10	0.1	0	[0.0, 0.1]	0

technique is widely used in time series problems [27–29], which divides time series into three parts: seasonal, trend, and residual components. The regional PM2.5 concentration variation is cyclical; thus, it is reasonable to adopt this preprocessing method. Finally, we select the PM2.5 data from five Chinese cities in the period from January 1, 2010, to December 31, 2015. To demonstrate the effectiveness of DSMS, it is compared with four training algorithms. It is also compared with seven representative prediction models. The experimental results show that the DSMS algorithm in the prediction model is effective, and the DS-DNM model has the excellent performance in the PM2.5 concentration prediction problem.

The main contributions of this paper are as follows:

1. A DS-DNM model for PM2.5 time series prediction is proposed. It is constructed in a hybrid manner. Experiments prove that it is more competitive than other models.
2. A new algorithm DSMS is proposed. The prediction model using it performs better.
3. A data preprocessing technique STL-LOESS is used in the DS-DNM model, which processes the PM2.5 data well.
4. This paper shows that DNM has the huge potential for the prediction problem and provides a future research direction.

The rest of this paper is organized as follows. Section 2 describes the original DNM model. Section 3 presents the proposed DS-DNM model. Section 4 shows experimental results of the proposed algorithm, four training algorithms, and seven prediction models. Section 5 summarizes this paper and gives future work.

2 Dendritic neuron model

Neural networks have evolved since their introduction and have succeeded in different fields such as image recognition [30], natural language processing [31, 32], and data prediction [33–35]. DNM is a dendritic neuron model, which simulates a biological neural network with a four-layer structure. It contains synaptic layer, dendrite layer, membrane layer, and soma layer. Figure 1 shows its morphological architecture. The DNM model is simple and efficient. It has the good performance in solving nonlinear problems. Its four-layer structure is described below.

2.1 Synaptic layer

The prominent presynapse receives information from the output of the previous neuron or the outside. The received information is expressed as $\{x_1, x_2, \ldots, x_i, \ldots, x_n\}$. Under the influence of different ions, the receptors at the synapse will appear in different states of excitation or inhibition, calculated as follows:

$$Y_{ij} = \frac{1}{1 + e^{-k\left(w_{ij}x_i - q_{ij}\right)}} \tag{1}$$

where Y_{ij} denotes the potential at the postsynaptic cells of the jth $(j = 1, 2, \ldots, M)$ dendrites from the ith $(i = 1, 2, \ldots, N)$ presynaptic axon terminal. k is a constant. Weight w_{ij} and threshold q_{ij} are generated by the training algorithm.

2.2 Dendrite layer

The dendritic layer performs a cumulative multiplication operation on the output of the salient layer. It is worth mentioning that when 0 and 1 are in the output of the synaptic layer, the computation of the dendrite layer can be regarded as the logic 'AND' operation. The dendrite layer is expressed as

$$Z_j = \prod_{i=1}^{N} Y_{ij} \tag{2}$$

2.3 Membrane layer

The resultant dendritic processed data are input to the membrane layer, which performs the cumulative calculation. Similar to the logical 'AND' operation in the dendrite

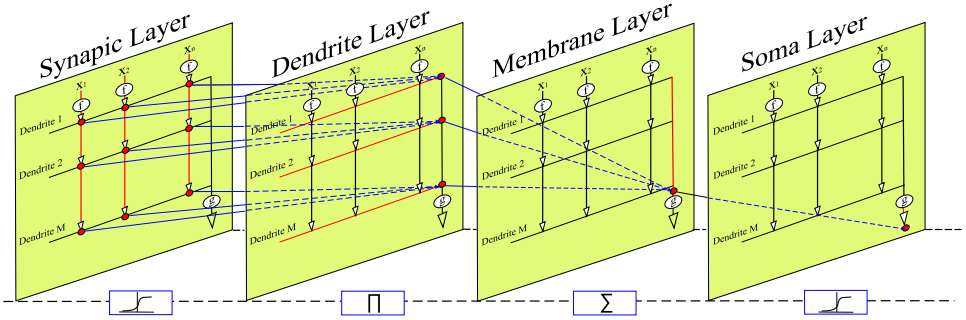

Fig. 1 Morphological architecture of dendritic neuron model

layer, the membrane layer has a logical 'OR' operation. The membrane layer is expressed as

$$V = \sum_{j=1}^{M} \prod_{i=1}^{N} Y_{ij} \tag{3}$$

2.4 Soma layer

The data are finally passed through the soma layer to determine whether the neuron is excited or not. This layer uses the sigmoid function to calculate the output as follows:

$$O = \frac{1}{1 + e^{-k(V-\theta)}} \tag{4}$$

where k is a constant and the threshold θ varies from 0 to 1.

3 Proposed DS-DNM model

In DNM model, parameters have the great impact. Back propagation (BP) is used in the original DNM model to determine weights w and thresholds q. However, the BP algorithm tends to fall into the local optimum, which prevents the DNM model from performing better. Therefore, we need an algorithm with powerful search capability to train the model. SMS has the notable performance for training the DNM model in classification problems. Inspired by it, we propose a novel algorithm based on SMS called DSMS.

3.1 States of matter search algorithm

SMS is a nature-inspired algorithm. It is based on the mechanism of thermal energy movement, which simulates individuals as interacting molecules. Its superior performance is obtained through different designs of exploration and exploitation ratios in three states of matter. The evolutionary process transitions from a gaseous state that exhibits a large number of collisions, to a relatively stable liquid state, and finally evolves to a completely stable solid state. The individual obtains the balance between exploration and exploitation during the evolution of simulated molecules. The evolutionary process of SMS is described below.

Direction vector

In the population X, the position of each individual X_i in the iterative process is controlled by the direction vector d_i. As the iterative process progresses, the individuals keep moving closer to the current optimal individual's position. The direction vector d_i is calculated as follows:

$$d_i^{k+1} = 0.5 \cdot d_i^k \cdot \left(1 - \frac{\text{nFES}}{\text{FES}}\right) + \frac{X_{\text{best}} - X_i}{\|X_{\text{best}} - X_i\|} \tag{5}$$

where FES is the maximum number of iterations, nFES is the current number of iterations, and X_{best} is the current optimal individual's position. The velocity vector v_i is calculated as

$$v_i = d_i \cdot \frac{\sum_{j=1}^{n} \left(b_j^h - b_j^l\right)}{n} \cdot \beta \tag{6}$$

where b_j^h and b_j^l are the upper and lower bounds of the *jth* individual, respectively. $\beta \in [0,\ 1]$. Through the updated direction vector and velocity vector, the new position is calculated as

$$X_{i,j}^{k+1} = X_{i,j}^k + v_{i,j} \cdot \lambda \cdot \left(b_j^h - b_j^l\right) \tag{7}$$

where $\lambda \in [0,\ 1]$.

Collisions

Collision behavior occurs during the evolution of molecules. When the distance between two molecules d_a and d_b is less than the collision distance, they exchange direction vectors, i.e., $d_a = d_b,\ d_b = d_a$. The collision distance is calculated as follows:

$$r = \frac{\sum_{j=1}^{n} \left(b_j^h - b_j^l\right)}{n} \cdot \eta \tag{8}$$

where $\eta \in [0,\ 1]$.

Random positions

There is a certain probability of random changes in the molecular evolution. To simulate this mechanism, the molecular positions are defined as follows:

$$X_{i,j}^{k+1} = \begin{cases} b_j^l + \text{rand}(0,\ 1) \cdot \left(b_j^h - b_j^l\right), & \text{if } s < H \\ X_{i,j}^{k+1}, & \text{if } s \geq H \end{cases} \tag{9}$$

where random number s is generated within the range [0, 1].

3.2 DSMS based on population selection strategy

Based on the population selection strategy, this paper proposes a novel algorithm DSMS. Inspired by differential evolution algorithm and its various variants in DNM model, DSMS divides the population into two sub-populations according to the individuals' ranking. During the iterative process, the worse subpopulation is updated by two strategies, which obtains the better balance between exploration and exploitation.

When the collision step in DSMS is completed, the fitness is calculated; a new population $X^r(k)$ is obtained by sorting the fitness from smallest to largest. The first half of the population size $\left[\frac{N}{2}\right]$ and the second half of $X^r(k)$ are defined as $X^g(k)$ and $X^b(k)$, respectively.

$$\begin{cases} X_i^g(k) = X_s^r(k) \\ X_i^b(k) = X_q^r(k) \end{cases} \tag{10}$$

where $s = 1,\ 2,\ \ldots,\ \left[\frac{N}{2}\right]$, $q = \left[\frac{N}{2}\right] + 1,\ \left[\frac{N}{2}\right] + 2,\ \ldots,\ N$.

Then, we update some individuals in the poor subpopulation via those in the optimal subpopulation. The first N/4 individuals are retained in $X^g(k)$, while a new individual is generated by every four individuals. Thus, a optimal subpopulation $G^g(k)$ is generated with the N/2 population size.

$$G_i^g(k) = \begin{cases} X_i^g(k), & 0 < i \le \frac{N}{4} \\ 0.25 \cdot \left[X_l^r(k) + X_{l+1}^r(k) + X_{l+2}^r(k) + X_{l+3}^r(k)\right], & \frac{N}{4} < i \le \frac{N}{2} \end{cases} \tag{11}$$

where $l \in \left[\frac{N}{4} + 1,\ \frac{N}{2}\right]$. The probability p is set to ensure that the optimal subpopulation $G(k)$ is randomly selected from the overall population with the $(1-p)$ probability:

$$G_i(k) = \begin{cases} G_i^g(k), & r \le p \\ X_r(k), & r > p \end{cases} \tag{12}$$

$$p = \frac{\text{nFES}}{\text{FES}} \tag{13}$$

where $i = 1,\ 2,\ \ldots,\ \left[\frac{N}{2}\right]$, $r \in [1,\ N]$. At the beginning, the optimal subpopulation $G(k)$ is randomly selected. As the number of iterations increases, $G(k)$ tends to select the excellent individuals after sorting. It makes the individuals more random when they are in the gaseous state. The population tends to search around the excellent individuals after gradually changing to the solid state.

The mutation operation is performed for the optimal subpopulation $G(k)$ to obtain the subpopulation $H(k)$. Then, a new subpopulation $V(k)$ is generated by subpopulations $H(k)$ and $X^b(k)$. The formula is as follows:

$$H_i(k) = G_{r1}(k) + F \cdot (G_{r2}(k) - G_{r3}(k)) \tag{14}$$

$$V_{i,j}(k) = \begin{cases} H_{i,j}(k), & \text{rand}(0,\ 1) \le CR \text{ or } j = j_{\text{rand}} \\ X_{i,j}^b(k), & \text{otherwise} \end{cases} \tag{15}$$

where $i = 1,\ 2,\ \ldots,\ \left[\frac{N}{2}\right]$, $j = 1,\ 2, \ldots,\ D$, $r \in [1,\ N]$. $r1$, $r2$, $r3$ are random integers with values ranging from 1 to $N/2$, respectively. The greedy selection strategy is used to select the better individuals compared with the original subpopulation $X^b(k)$. The formula is as follows:

$$X_i^b(k+1) = \begin{cases} V_i(k), & \text{if } (V_i(k)) < f\left(X_i^b(k)\right) \\ X_i^b(k), & \text{if } (V_i(k)) \ge f\left(X_i^b(k)\right) \end{cases} \tag{16}$$

Figure 2 shows the evolutionary process of DSMS. From it, the population gradually evolves from gaseous to solid states, and the evolutionary strategy of individuals gradually shifts from stochastic to convergent. Meanwhile, during the evolution of population, the poor subpopulation is updated. Based on the probability parameter p, which varies with the number of iterations, a majority of updated target samples are random individuals in the gaseous state. Then, random and superior individuals coexist in the liquid state. Finally, when the population becomes solid, the poor subpopulation is approximately dependent on the superior subpopulation. Thus, DSMS relies on the overall state change and the choice of individual variation strategies in local areas.

3.3 Seasonal-trend decomposition based on locally weighted scatterplot smoothing

Seasonal-trend decomposition based on locally weighted scatterplot smoothing (STL-LOESS) is an algorithm for the temporal decomposition, which has been widely used in finance, medicine, sociology, oceanography[29], and other fields [21, 36, 37]. Through STL-LOESS processing, the raw data are divided into three parts as follows:

$$Y_t = S_t + T_t + R_t \tag{17}$$

where Y_t, S_t, T_t, R_t correspond to original time series data, seasonal component, trend component, and residual component, respectively. Figure 3 shows the original PM2.5 data in Shenyang from May 1, 2013, to March 13, 2015. The seasonal component, trend component, and residual component are processed by STL-LOESS. The STL-LOESS decomposition process is described below.

3.3.1 Loess

In loess, seasonal and trend components need to be smoothed. x is the current independent variable (in this paper represents time), and $g(x)$ is the locally weighted regression curve. To calculate $g(x)$, a positive integer q ($q \le n$) is selected. The q values closest to x are selected to assign an adjacency weight W according to their distance from x. The selected values are $(x_i,\ y_i)$, $i = 1, 2, \ldots, q$. x_i are the independent variables (time) and y_i are the dependent variables (concentration of PM2.5). W is calculated as follows:

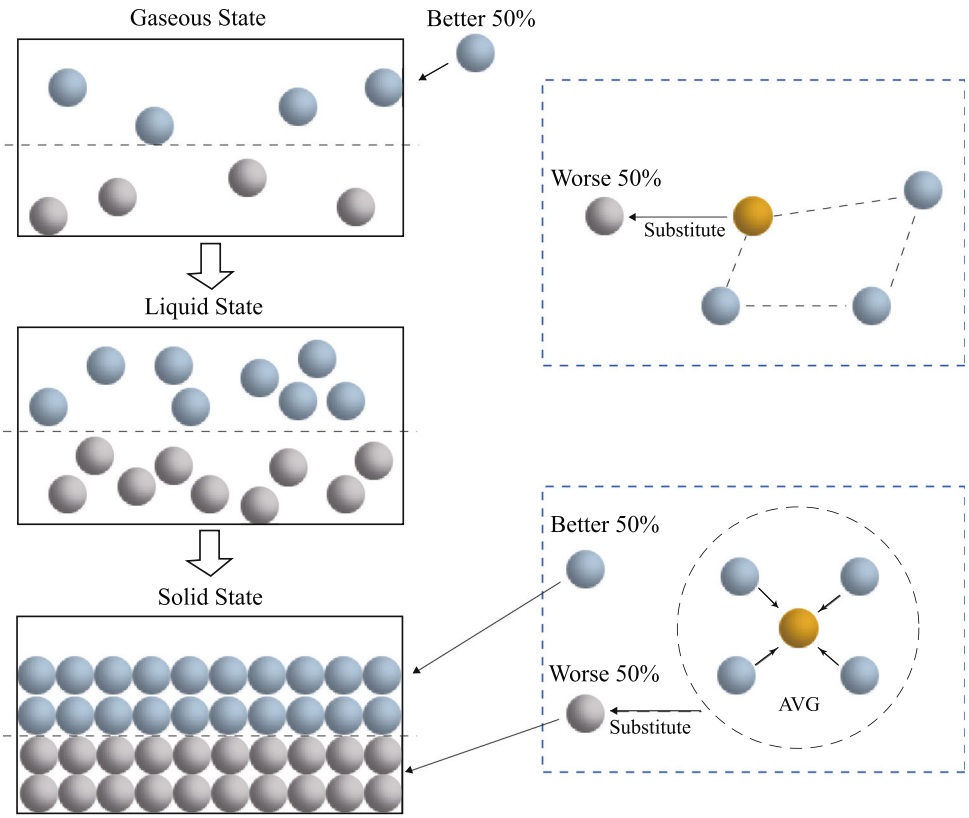

Fig. 2 Evolutionary process of the DSMS algorithm

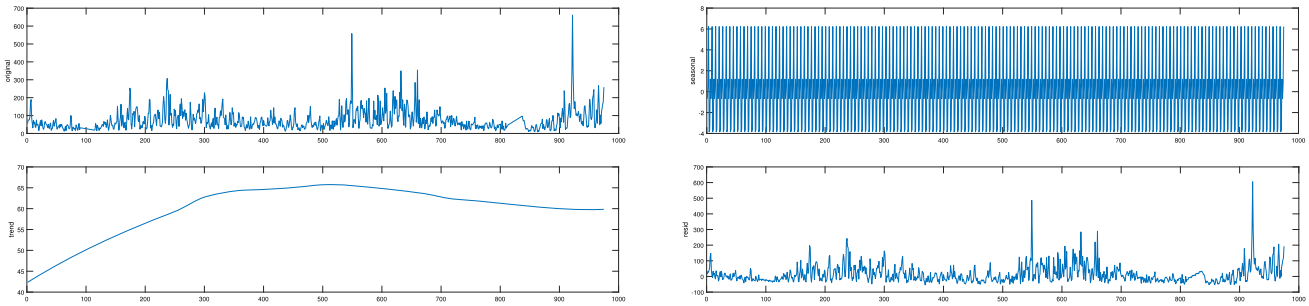

Fig. 3 Breakdown of PM2.5 data in Shenyang

$$W(k) = \begin{cases} (1-k^3)^3, & \text{if } 0 \le k < 1 \\ 0, & \text{if } k \ge 1 \end{cases} \tag{18}$$

The neighborhood weights of x_i are expressed as:

$$U_i x = W\frac{\|x_i - x\|}{\lambda_q x}. \tag{19}$$

where $\lambda_q(x)$ indicates the farthest distance from x among the selected q values. It can be seen that the neighborhood weights are decreasing as the distance from x to x_i increases. Similarly, data with weights U_i need to be fitted when $q > n$. Define the distance $\lambda_q(x)$ from x to the farthest x_i as follows:

$$\lambda_q(x) = \lambda_n(x)\frac{q}{n} \tag{20}$$

where $\lambda_n(x)$ is the distance from x to the farthest x_i. As q keeps increasing, the locally weighted regression curve $g(x)$ will become smooth. As q tends to infinity, the weights U_i converge to 1 and $g(x)$ converge to a general least squares polynomial of order d.

3.3.2 Decomposition process

There are two nested loops, i.e., inner and outer loops, in the STL-LOESS decomposition process. In the inner loop, seasonal and trend terms are updated. In the outer loop,

robustness weights are generated based on the number of residuals. The specific steps are as follows:

- Step 1: Detrending. Minus the trend component of the previous round's results, i.e., $Y_u - T_u^{(k)}$.
- Step 2: Cycle-subseries smoothing. Regress the series with loess to obtain the temporary seasonal series $C_u^{(k+1)}$.
- Step 3: Low-pass filtering. The sequence $C_u^{(k+1)}$ is processed by moving average and loess regression to obtain the sequence $L_u^{(k+1)}$.
- Step 4: Detrending of smoothed cycle-subseries, i.e., $S_u^{(k+1)} = C_u^{(k+1)} - L_u^{(k+1)}$.
- Step 5: Deseasonalizing. Removal of periodic quantities from a sequence, i.e., $Y_u - S_u^{(k+1)}$.
- Step 6: Trend smoothing. The obtained series $S_u^{(k+1)}$ is processed again using the loess regression to obtain the trend value $T_u^{(k)}$.

3.3.3 Parameter selection

To obtain satisfactory decomposition results, it is necessary to choose the appropriate STL-LOESS parameters. From previous studies, we experimentally validate the obtained parameters. PM2.5 concentration has the periodic change, but it is affected by the geography of different regions. According to the experimental study [38], the period n_p is set to 7. The regression data component n_f in loess is 0.6. The number of residuals n_i to be re-weighted is 3.

3.4 DS-DNM

This section describes the details of combining STL-LOESS and DNM in DS-DNM model. It also describes how the DSMS algorithm is used to train the DNM model. Figure 4 shows the flow chart of DS-DNM. STL-LOESS preprocesses the data. It decomposes the original data into three components: seasonal component, trend component, and residual component.

3.4.1 Treatment of seasonal, trend, and residual components

Since PM2.5 concentration change is periodic and the periodicity is generally stable, the seasonal component keeps constant. A trend component can be regarded as a continuous function that changes with time; thus, a cubic polynomial is used to fit it as follows:

$$T_t = at^3 + bt^2 + ct + d \tag{21}$$

where parameters a, b, c, and d are determined by the least square estimation method.

The residual component is divided into two parts, i.e., training set and test set. It is input to DNM model for training. For the input, the one-dimensional time series is put into the high dimension, and the concentration data of two consecutive days are used to predict the next day. The input vector T and the target vector V are structured as follows:

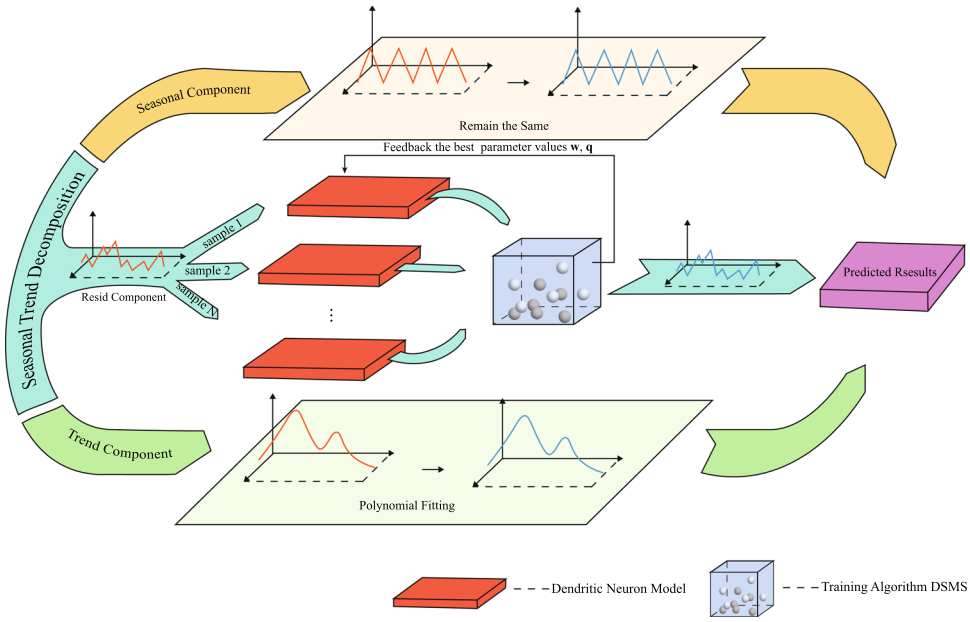

Fig. 4 Scheme of DS-DNM

$$T_r = \begin{bmatrix} x^{(1)} & x^{(2)} & x^{(3)} & \dots & x^{(N-2)} \\ x^{(2)} & x^{(3)} & x^{(4)} & \dots & x^{(N-1)} \end{bmatrix} \tag{22}$$

$$V_r = \begin{bmatrix} x^{(3)} & x^{(4)} & x^{(5)} & \dots & x^{(N)} \end{bmatrix} \tag{23}$$

where N is the sequence length.

3.4.2 DSMS for training DNM model

DSMS randomly generates the initial population X where the individual is $X_i^{(k)} = \{w_{11}, w_{12}, \ldots, w_{MN}, q_{11}, q_{11}, \ldots, q_{MN}\}$. DSMS iteratively updates the processed residuals as the training and test sets to find the optimal solution. Fitness of the algorithm during training is expressed as the error between the target value and the predicted value, which is calculated as:

$$E = (T_r - V_r)^2 \tag{24}$$

3.4.3 Predictive data acquisition

The optimal parameters w and q obtained by DSMS are used in DS-DNM model. The obtained high-dimensional data are reduced to a one-dimensional series by an inverse processing. We obtain the final predicted value T_p via adding the residual value T_r with the trend value T_t and the period value T_s:

$$T_p = T_r + T_t + T_s \tag{25}$$

4 Experimental results

The experimental data are a dataset of daily average PM2.5 concentrations over time for five Chinese cities including Beijing, Shanghai, Guangzhou, Chengdu, and Shenyang, which is published by the US Consulate in China.

Algorithm 1 DSMS for training DNM mode

```
Initialization: parameters are set as shown in Table 1. nFES ← 0.
Generate the initial population H.
while nFES ≤ FES do
    for i = 1 to N do
        Update d_i and v_i based on Eq. (5) - (6).
        Update p_i based on Eq. (7).
        Update r and rr based on Eq. (8).
        if rr < r then
            Exchange direction vectors.
        end if
        Update p_i^(k) randomly based on Eq. (9).
        Calculate V ← p_i^(k) (via DNM model).
        Calculate E based on Eq. (24).
        Sort fitness values.
        Obtain X_i^g(k) and X_i^b(k) based on Eq. (10).
        Update G_i^g(k) and G_i(k) based on Eqs. (11) - (13).
        Update V_{i,j}(k) based on Eqs. (14) - (15).
        Renew poor subpopulations via Eqs. (16).
        Parameters k and qs are extracted according to the current optimal
    solution.
    end for
end while
```

The pseudo-code of DSMS training DNM model is shown in Algorithm 1. E in the model is used as the fitness. The smaller E is, the more effective the DSMS is.

4.1 Description of dataset

We experimentally select PM2.5 monitoring data from the US Consulate in China for five Chinese cities over a six-year period, i.e., from January 1, 2010, to December 31,

Table 2 Description of PM2.5 test data in five cities

Dataset	Training data Year/month	Test data Year/month	Amount of data Days
PM2.5 of Beijing	2010/1/1–2014/3/14	2014/3/15–2015/12/31	2190
PM2.5 of Shenyang	2013/5/1–2015/3/13	2015/3/14–2015/12/31	974
PM2.5 of Chengdu	2012/6/1–2014/12/3	2014/12/4–2015/12/31	1308
PM2.5 of Shanghai	2012/1/1–2014/10/19	2014/10/20–2015/12/31	1460
PM2.5 of Guangzhou	2012/1/1–2014/10/19	2014/10/20–2015/12/31	1460

2015. The data are daily average PM2.5 concentrations with some missing data caused by environmental or technical reasons and are filled by linear interpolation. For each data, the first 70% is set as the training set and the last 30% as the test set. The experimental data are described in Table 2.

4.2 Evaluation criteria

In order to scientifically and accurately assess the model prediction, three evaluation indexes consisting of mean square error (MSE), mean absolute percentage error (MAPE), and mean absolute error (MAE) are used. Their formulas are shown as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(O_i - T_i)^2 \tag{26}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left\|\frac{T_i - O_i}{T_i}\right\| \tag{27}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\|T_i - O_i\| \tag{28}$$

4.3 Parameter settings

Three parameters need to be determined in the DS-DNM model. They are the constant k of the sigmoid function in the synaptic layer, the threshold qs of the soma layer, and the number of dendrite layers M. Different parameter settings affect the performance of the model. To determine which values are the most suitable for the model, we test 25 combinations of the three parameters using Genichi Taguchi's orthogonal experiment. For the test, each datum in five cities is run 20 times. The population size and number of iterations are set to 100 and 30,000, respectively. Table 3 shows 25 groups of parameter settings. Table 4 shows the ranking of 25 groups of parameter settings in each city according to MSE. The experimental results show that the DS-DNM model works best when $k = 5$, $qs = 1$, and $M = 10$.

4.4 Normalization

To reduce the complexity of the data, the original data are normalized into $[0, 1]$. The normalization formula is as follows:

$$x_i^{(n)} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{29}$$

where $x_{\max}$ and $x_{\min}$ are maximum and minimum values, respectively. When the predicted data are output, a reverse normalization operation is required to bring the result back into the original value.

Table 3 Parameter settings in the orthogonal experiment

Parameter	k	qs	M	Parameter	k	qs	M	Parameter	k	qs	M
Group 1	1	0.1	2	Group 10	5	1	10	Group 19	15	0.7	15
Group 2	1	0.3	10	Group 11	10	0.1	15	Group 20	15	1	2
Group 3	1	0.5	20	Group 12	10	0.3	2	Group 21	20	0.1	5
Group 4	1	0.7	5	Group 13	10	0.5	10	Group 22	20	0.3	15
Group 5	1	1	15	Group 14	10	0.7	20	Group 23	20	0.5	2
Group 6	5	0.1	20	Group 15	10	1	2	Group 24	20	0.7	10
Group 7	5	0.3	5	Group 16	15	0.1	10	Group 25	20	1	20
Group 8	5	0.5	15	Group 17	15	0.3	20				
Group 9	5	0.7	2	Group 18	15	0.5	5				

Table 4 Ranking results in the orthogonal experiment

Group index	Beijing	Shenyang	Chengdu	Shanghai	Guangzhou	Overall ranking
Group 1	25	25	25	25	25	25
Group 2	24	24	24	24	24	24
Group 3	22	23	22	23	22	23
Group 4	21	21	21	21	21	21
Group 5	19	20	20	19	19	20
Group 6	23	22	23	22	23	22
Group 7	17	18	17	18	17	18
Group 8	10	13	8	12	8	11
Group 9	7	15	3	7	7	10
Group 10	1	1	2	2	3	1
Group 11	20	19	19	20	20	19
Group 12	12	11	15	14	15	15
Group 13	11	12	11	6	11	9
Group 14	2	8	2	3	6	5
Group 15	15	1	6	13	4	3
Group 16	18	17	18	17	18	17
Group 17	8	10	16	10	13	13
Group 18	1	7	10	8	10	7
Group 19	14	5	4	9	9	6
Group 20	13	14	5	2	1	8
Group 21	16	16	12	15	16	16
Group 22	9	9	14	11	12	12
Group 23	5	6	9	16	14	14
Group 24	4	3	7	5	5	2
Group 25	3	4	13	1	2	4

Table 5 MSE comparison of training algorithms

	MSE					
	Beijing	Shenyang	Chengdu	Shanghai	Guangzhou	Total
DSMS	883.38162	818.814185	71.86893	110.13355	27.589105	5
DPDE	1194.331898	2060.767031	78.65910962	200.5661876	69.16750645	0
SCJADE	4843.150555	4930.175435	8476.12182	7392.928476	53.61074971	0
BBO	1333.31963	939.5004343	79.89269988	157.678848	35.49607347	0
BP	1142.611843	920.6199894	83.04899327	175.0730765	48.38879275	0

Table 6 MAPE comparison of training algorithms

	MAPE					
	Beijing	Shenyang	Chengdu	Shanghai	Guangzhou	Total
DSMS	0.000279643	**0.00042833**	**0.000120303**	**0.000211008**	**0.000116485**	**4**
DPDE	**0.0002776**	0.000619494	0.000123371	0.000227731	0.000157457	1
SCJADE	0.00040887	0.001021096	0.001595116	0.001459852	0.000257467	0
BBO	0.00029717	0.000561419	0.000129213	0.000229094	0.000087215	0
BP	0.000281931	0.00055626	0.000134764	0.000252546	0.000140428	0

Table 7 MAE comparison of training algorithms

	MAE					
	Beijing	Shenyang	Chengdu	Shanghai	Guangzhou	Total
DSMS	**20.07915**	18.605325	**5.22165**	**7.67925**	**3.906435**	4
DPDE	23.98902621	25.7977413	5.830522242	9.194371287	6.022538918	0
SCJADE	56.50916674	45.67571775	59.93775912	57.10974117	6.118918113	0
BBO	25.46471521	**17.14728116**	5.40840392	8.194505857	4.094493923	1
BP	26.4275625	22.57772662	7.288425303	10.16802185	5.67465137	0

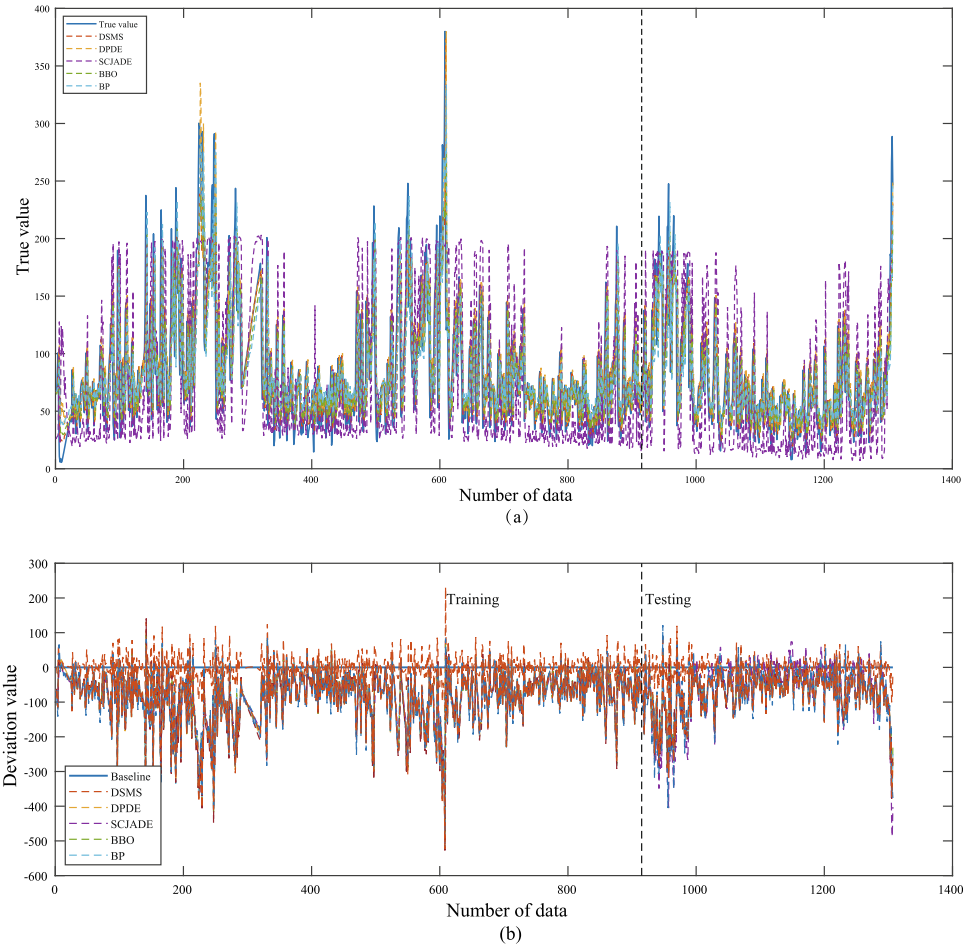

Fig. 5 Prediction and deviation plots of models using different training algorithms on Chengdu data

4.5 Performance comparison of training algorithms

In this section, we compare the training effect of DSMS with three optimization algorithms, including directional permutation differential evolution algorithm (DPDE) [39], chaotic differential evolution algorithm based on roulette wheel selection (SCJADE) [40], and biogeography-based optimization (BBO) [41]. It is also compared with the training algorithm BP in the original DNM model [42]. The initial parameters of the comparison algorithms are obtained from the above-mentioned literatures. For a fair comparison, the most appropriate parameters need to be set. Due to the powerful and fast convergence characteristics of DSMS, the best results are obtained when the number of iterations is set to 3000. Otherwise, the model is prone to fall into overfitting. For other algorithms, the best results are obtained when the number of iterations is set to 10,000. Five algorithms are run 30 times to train the DNM model and predict PM2.5 concentrations in five cities. The results are shown in Tables 5, 6, and 7 where the best value is highlighted in boldface.

According to Tables 5, 6, and 7, DSMS has the least MSE, MAPE, and MAE values in comparison with the other algorithms. Two exceptions are that DPDE has the least MAPE value in Beijing and BBO has the least MAE

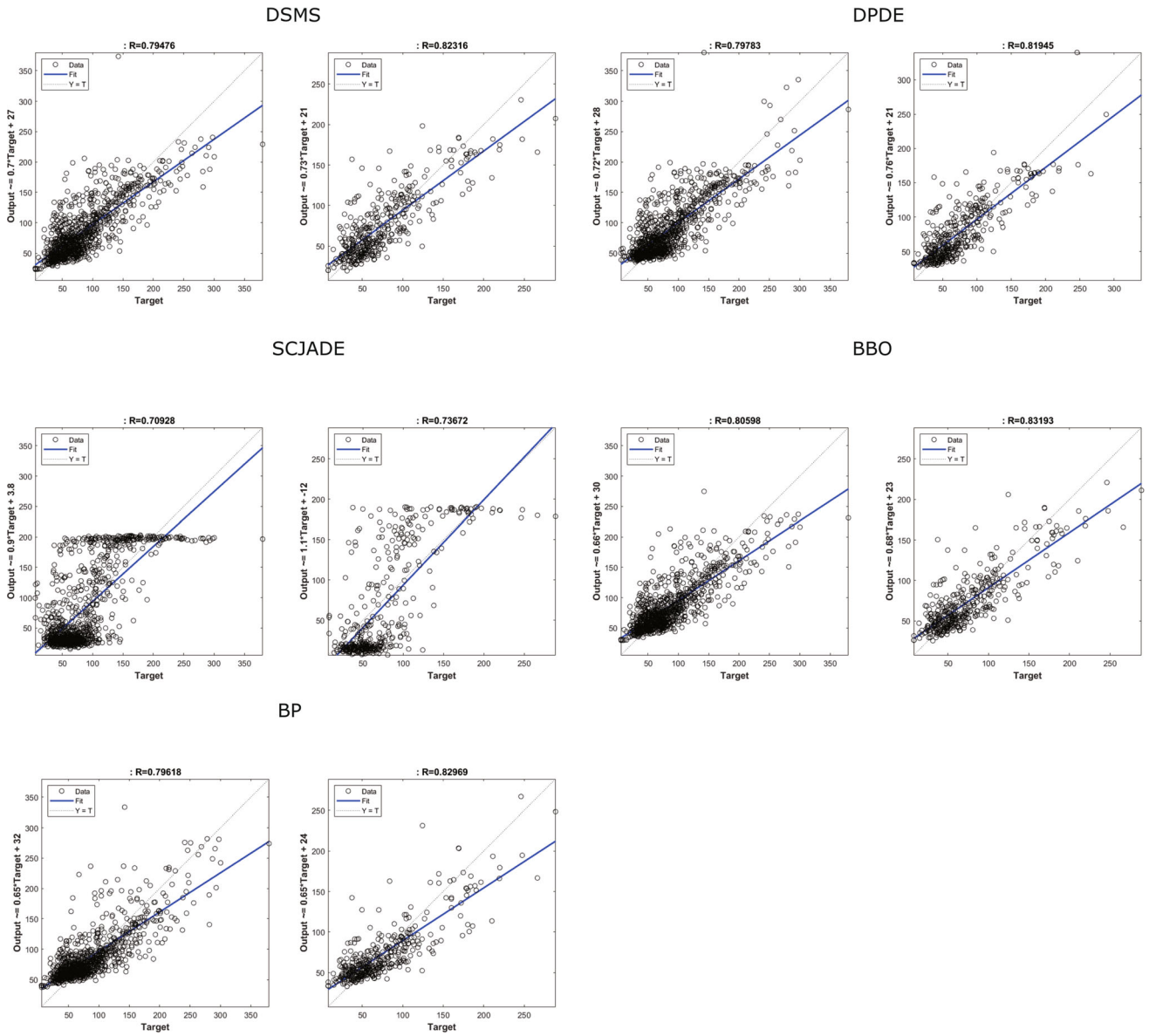

Fig. 6 Correlation coefficient graphs based on different training algorithms

Table 8 MSE comparison of eight prediction models

	MSE					
	Beijing	Shenyang	Chengdu	Shanghai	Guangzhou	Total
DS-DNM	**883.38162**	**818.814185**	**71.86893**	**110.13355**	**27.589105**	**5**
Elman	1214.49942	991.27306	115.63718	205.85143	54.32243	0
SVM_{RBF}	5547.82376	3639.89525	2028.10819	977.59598	612.23797	0
SVM_{LF}	3565.46440	1951.15810	674.67220	796.06690	294.64700	0
ANFIS	3730.80416	2125.14289	647.14104	732.38333	293.24394	0
MLP	1853.25486	1721.61619	221.19380	262.76335	79.73914	0
LSTM	6106.91294	3101.01687	1327.10840	1512.23015	488.77707	0
SDNN	1105.24809	911.777895	99.85854	192.39068	62.9107	0

Table 9 MAPE comparison of eight prediction models

	MAPE					
	Beijing	Shenyang	Chengdu	Shanghai	Guangzhou	Total
DS-DNM	0.000279643	**0.00042833**	0.000120303	**0.000211008**	0.000116485	**2**
Elman	**0.000216455**	0.000668442	0.000086628	0.000263272	**0.000070663**	2
SVM_{RBF}	0.00051056	0.001305566	0.000438818	0.000591614	0.000144555	0
SVM_{LF}	0.00041322	0.0010672	0.00010877	0.00057366	0.00028181	0
ANFIS	0.000379478	0.000887179	0.00018009	0.000399773	0.000148908	0
MLP	0.000241004	0.000714427	**0.00007629**	0.000247314	0.000062398	1
LSTM	0.000303425	0.000865391	0.000245949	0.000310894	0.000167328	0
SDNN	0.000264431	0.000482369	0.000110788	0.000283405	0.000158342	0

Table 10 MAE comparison of eight prediction models

	MAE					
	Beijing	Shenyang	Chengdu	Shanghai	Guangzhou	Total
DS-DNM	**20.07915**	18.605325	**5.22165**	**7.67925**	**3.906435**	**4**
Elman	27.02807097	**18.38115494**	8.274835585	10.41369438	5.700727668	1
SVM_{RBF}	56.62420459	37.52533748	35.27241399	22.54283337	19.24984806	0
SVM_{LF}	41.3582	24.6066	18.7175	28.2147	12.0583	0
ANFIS	43.59985604	25.23860845	18.50936716	18.9818359	12.42712893	0
MLP	31.29095547	20.76986178	10.73036377	10.42384863	6.238967666	0
LSTM	52.80783242	32.68543188	24.91178899	25.67223136	15.61724182	0
SDNN	25.75872	22.790445	7.74337	10.99625	6.527765	0

Table 11 Friedman test on all models

Model	Ranking of MSE	Model	Ranking of MAPE	Model	Ranking of MAE
DS-DNM	**1**	DS-DNM	5	**DS-DNM**	**1**
Elman	3	Elman	2	Elman	3
SVM_{RBF}	8	SVM_{RBF}	8	SVM_{RBF}	8
SVM_{LF}	6	SVM_{LF}	3	SVM_{LF}	6
ANFIS	5	ANFIS	6	ANFIS	5
MLP	4	**MLP**	**1**	MLP	4
LSTM	7	LSTM	7	LSTM	7
SDNN	2	SDNN	4	SDNN	2

value in Shenyang. Results show that DSMS outperforms DPDE, SCJADE, BBO, and BP in training the DNM model for PM2.5 concentrations prediction. Thus, DSMS is an effective and competitive algorithm.

Figure 5 shows the prediction results of the model with different training algorithms. Figure 7a shows the predicted and true values of PM2.5 in Chengdu. Figure 5b is obtained by subtracting the predicted and true values. The baseline indicates that the predicted and true values are same, which is the ideal state. From Fig. 7, around the average true value, all the algorithms have good results except for the SCJADE algorithm. Moreover, DSMS tends to achieve the best result when the true value is larger or smaller than the average value. Figure 6 shows the correlation coefficients of different algorithms. The calculated correlation coefficient R reflects the prediction ability of model with different algorithms. Except for the SCJADE algorithm, four prediction results are good. It reflects that DS-DNM model adopts STL-LOESS to preprocess the data well and the residual values are effectively handled.

4.6 Performance comparison with other prediction models

Seven prediction models in this section are used to compare with the DS-DNM model, including Elman network, two types of SVM models, ANFIS, MLP, LSTM, and SDNN. We choose a fuzzy c-means clustering as the

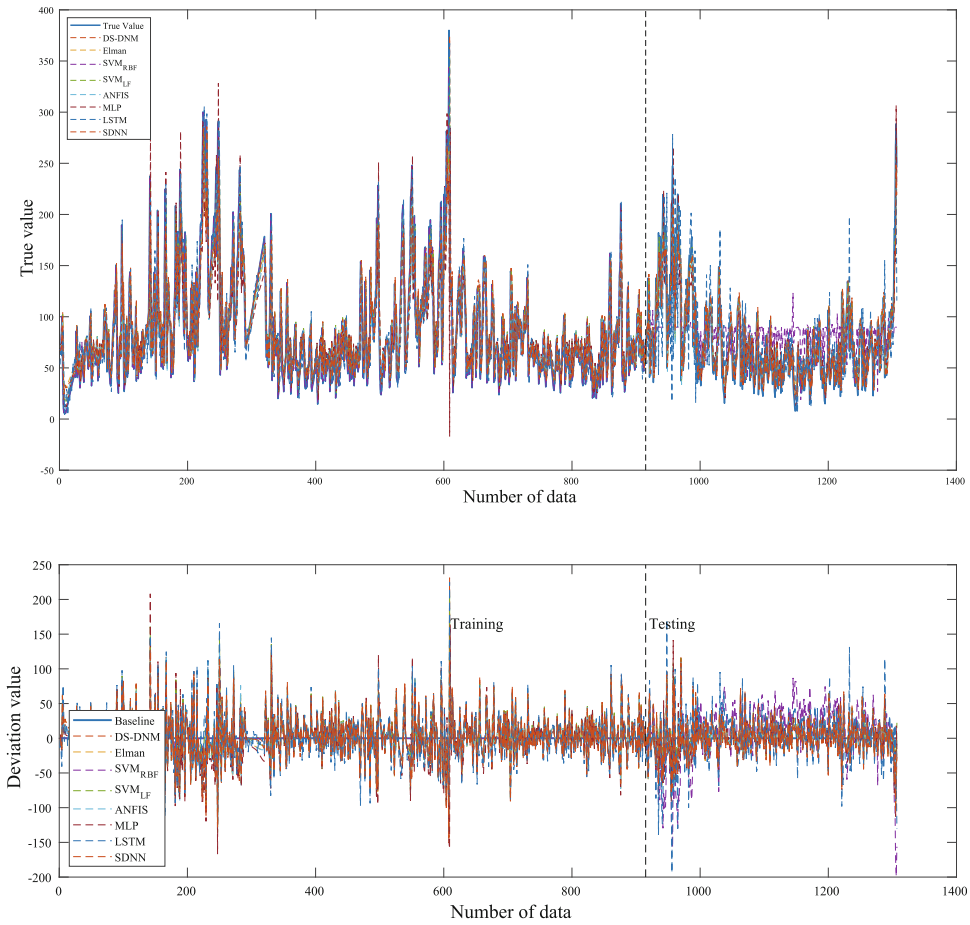

Fig. 7 Prediction and deviation plots using different prediction models on Chengdu data

generation method in ANFIS. Radial basis functions and linear functions are chosen as kernel functions in SVM, labeled as SVM_{RBF} and SVM_{LF}. The number of hidden layers in MLP, Elman, and LSTM models is 10, 15, and 60, respectively.

All eight prediction models run 30 times and the average value is calculated. Tables 8, 9, and 10 show the prediction results where the best value is highlighted in boldface. It can be seen from Tables 8 and 10 that DS-DNM has the best MSE and MAE values. However, in Table 9, Elman network has two best MAPE values and MLP has one. This is because the prediction of DS-DNM decreases when the actual values are biased towards local or global extremes, resulting in large MAPE values when the actual values are very small. To obtain a reliable conclusion, Table 11 shows the results of the Friedman test where the best model is highlighted in boldface. DS-DNM performs better than other models according to MSE and MAE, whereas MLP has the better performance in terms of MAPE. Figure 7 shows the prediction results of different prediction models. Figure 8 shows the correlation coefficient graphs based on different prediction models. From Figs. 7 and 8, all the prediction models show the good performance except for SVM and LSTM.

5 Conclusion

For the complexity of PM2.5 concentration time series, we propose the DS-DNM prediction model. A STL-LOESS data processing technique is used to decompose the complex time series into three parts: season, trend, and residual. It better identifies the internal characteristics of time series. An innovation-based population selection strategy is proposed to train the DS-DNM model. By decomposing the data and processing the structure in chunks, the model possesses the ability to accurately predict the air quality time series. The experiments are carried out on PM2.5 concentration data from five cities. Compared with four training algorithms and seven prediction models, the powerful prediction ability of DS-DNM model is verified.

However, the DS-DNM model also has some limitations. Its prediction ability decreases when the actual values are biased towards extreme values, and it is currently

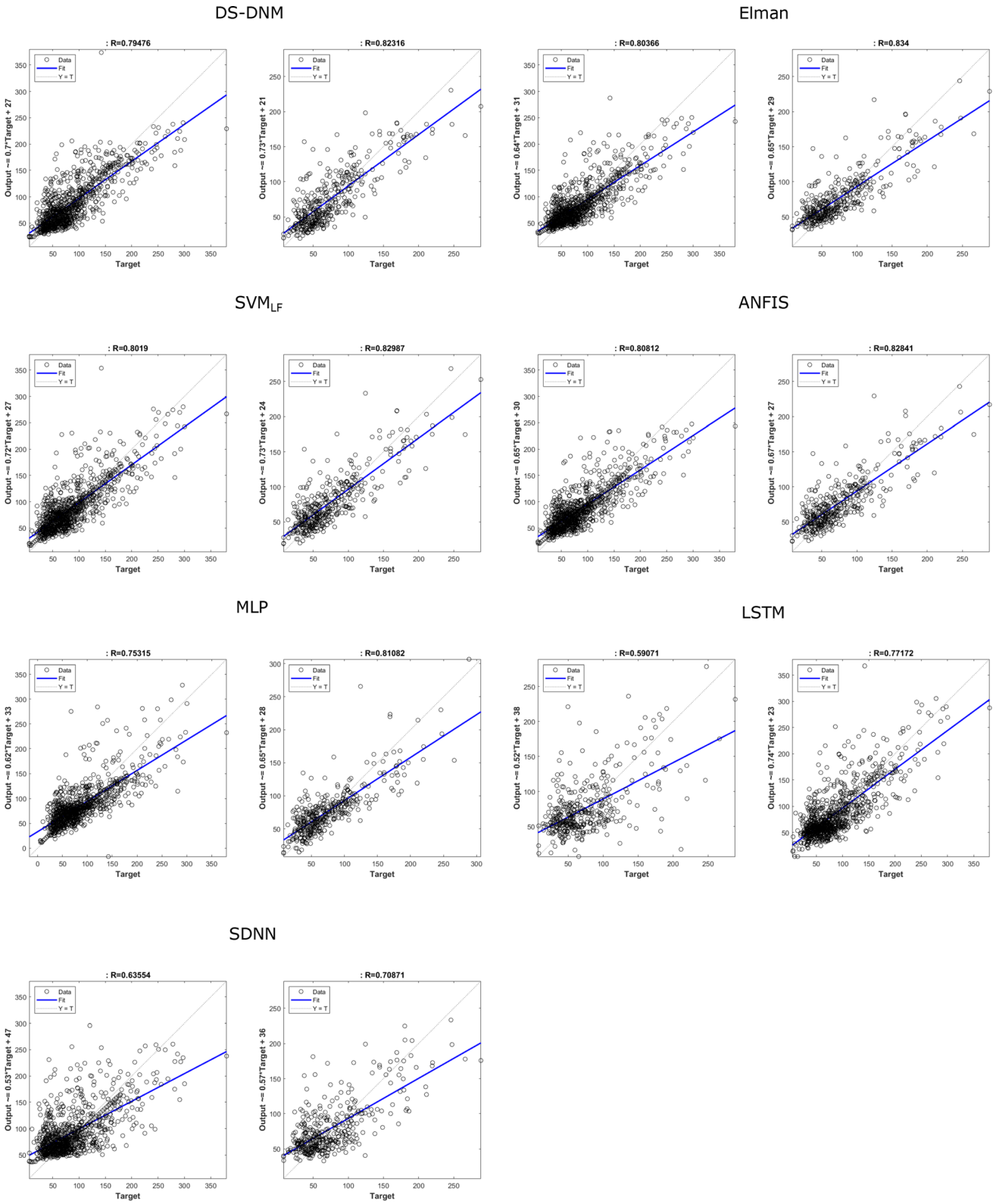

Fig. 8 Correlation coefficient graphs based on different prediction models

stuck on univariate time series problems. In the future, we will investigate the ability of DS-DNM model on multi-objective prediction problems and also explore the possibilities of the model in other prediction fields.

Acknowledgements This research was partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP22H03643 and JP19K22891, Japan Science and Technology Agency (JST) Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant JPMJSP2145, and JST through the Establishment of University Fellowships towards the Creation of Science Technology Innovation under Grant JPMJFS2115.

Author contributions ZY helped in conceptualization, writing—original draft, methodology, software. SG was involved in conceptualization, writing—review and editing, supervision, methodology, software. YW and LG helped in drawing. JL contributed to methodology and software. CH helped in drawing, software.

Funding This research was partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP22H03643 and JP19K22891, Japan Science and Technology Agency (JST) Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant JPMJSP2145, and JST through the Establishment of University Fellowships towards the Creation of Science Technology Innovation under Grant JPMJFS2115.

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest Title: Prediction of PM2.5 time series by seasonal trend decomposition-based dendritic neuron model. The authors whose names are listed immediately below certify that they have no affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge, or beliefs) in the subject matter or materials discussed in this manuscript. Author names: Zijing Yuan, Shangce Gao, Yirui Wang, Jiayi Li, Chunzhi Hou, and Lijun Guo.

Financial interests The authors declare they have no financial interests.

Non-financial interests None.

Author agreement statement We declare that this manuscript is original, has not been published before, and is not currently being considered for publication elsewhere. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us. We understand that the Corresponding Author is the sole contact for the Editorial process. He/she is responsible for communicating with the other authors about progress, submissions of revisions, and final approval of proofs. Author names: Zijing Yuan, Shangce Gao, Yirui Wang, Jiayi Li, Chunzhi Hou, and Lijun Guo.

Ethical approval No human or animal subjects were involved in this experiment.

Consent to participate No human subjects were involved in this experiment.

Consent to publish All authors approved the manuscript for publication.

References

1. Feng S, Gao D, Liao F, Zhou F, Wang X (2016) The health effects of ambient pm2. 5 and potential mechanisms. Ecotoxicol Environ Saf 128:67–74. https://doi.org/10.1016/j.jclepro.2017.02.092
2. Zhang H, Hu J, Qi Y, Li C, Chen J, Wang X, He J, Wang S, Hao J, Zhang L et al (2017) Emission characterization, environmental impact, and control measure of PM2. 5 emitted from agricultural crop residue burning in China. J Clean Prod 149:629–635. https://doi.org/10.1016/j.jclepro.2017.02.092
3. Ma J, Cheng JC, Lin C, Tan Y, Zhang J (2019) Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques. Atmos Environ 214:116885. https://doi.org/10.1016/j.atmosenv.2019.116885
4. Binkowski FS, Roselle SJ (2003) Models-3 community multiscale air quality (CMAQ) model aerosol component 1. Model description. J Geophys Res Atmos. https://doi.org/10.1029/2001JD001409
5. Schmidt H, Derognat C, Vautard R, Beekmann M (2001) A comparison of simulated and observed ozone mixing ratios for the summer of 1998 in Western Europe. Atmos Environ 35(36):6277–6297. https://doi.org/10.1016/S1352-2310(01)00451-4
6. Schaap M, Van Der Gon HD, Dentener F, Visschedijk A, Van Loon M, Ten Brink H, Putaud J-P, Guillaume B, Liousse C, Builtjes P (2003) Anthropogenic black carbon and fine aerosol distribution over Europe. J Geophys Res Atmos. https://doi.org/10.1029/2003JD004330
7. Duan W, Wang X, Cheng S, Wang R, Zhu J (2021) Influencing factors of $PM_{2.5}$ and O_3 from 2016 to 2020 based on DLNM and WRF-CMAQ. Environ Pollut 285:117512. https://doi.org/10.1016/j.envpol.2021.117512
8. Kukkonen J, Partanen L, Karppinen A, Ruuskanen J, Junninen H, Kolehmainen M, Niska H, Dorling S, Chatterton T, Foxall R et al (2003) Extensive evaluation of neural network models for the prediction of NO_2 and PM_{10} concentrations, compared with a deterministic modelling system and measurements in central Helsinki. Atmos Environ 37(32):4539–4550. https://doi.org/10.1016/S1352-2310(03)00583-1
9. Wu J, Li J, Peng J, Li W, Xu G, Dong C (2015) Applying land use regression model to estimate spatial variation of $PM_{2.5}$ in Beijing, China. Environ Sci Pollut Res 22(9):7045–7061. https://doi.org/10.1007/s11356-014-3893-5
10. Lei MT, Monjardino J, Mendes L, Gonçalves D, Ferreira F (2019) Macao air quality forecast using statistical methods. Air Qual Atmos Health 12(9):1049–1057. https://doi.org/10.1007/s11869-019-00721-9
11. Zhou Y, Chang F-J, Chang L-C, Kao I-F, Wang Y-S, Kang C-C (2019) Multi-output support vector machine for regional multi-step-ahead $PM_{2.5}$ forecasting. Sci Total Environ 651:230–240. https://doi.org/10.1016/j.scitotenv.2018.09.111
12. Huang C-J, Kuo P-H (2018) A deep CNN-LSTM model for particulate matter ($PM_{2.5}$) forecasting in smart cities. Sensors 18(7):2220. https://doi.org/10.3390/s18072220

13. McKendry IG (2002) Evaluation of artificial neural networks for fine particulate pollution (PM_{10} and $PM_{2.5}$) forecasting. J Air Waste Manag Assoc 52(9):1096–1101. https://doi.org/10.1080/10473289.2002.10470836
14. Wang Y, Gao S, Yu Y, Cai Z, Wang Z (2021) A gravitational search algorithm with hierarchy and distributed framework. Knowl Based Syst 218:106877. https://doi.org/10.1016/j.knosys.2021.106877
15. Zhan C, Jiang W, Lin F, Zhang S, Li B (2022) A decomposition-ensemble broad learning system for AQI forecasting. Neural Comput Appl 34(21):18461–18472. https://doi.org/10.1007/s00521-022-0744
16. Zhan C, Jiang W, Min H, Gao Y, Tse CK (2022) Human migration-based graph convolutional network for PM2.5 forecasting in post-COVID-19 pandemic age. Neural Comput Appl. https://doi.org/10.1007/s00521-022-0787
17. Zhao J, Deng F, Cai Y, Chen J (2019) Long short-term memory-Fully connected (LSTM-FC) neural network for $PM_{2.5}$ concentration prediction. Chemosphere 220:486–492. https://doi.org/10.1016/j.chemosphere.2018.12.128
18. Pérez P, Trier A, Reyes J (2000) Prediction of $PM_{2.5}$ concentrations several hours in advance using neural networks in Santiago, Chile. Atmos Environ 34(8):1189–1196. https://doi.org/10.1016/S1352-2310(99)00316-7
19. Qiao W, Tian W, Tian Y, Yang Q, Wang Y, Zhang J (2019) The forecasting of $PM_{2.5}$ using a hybrid model based on wavelet transform and an improved deep learning algorithm. IEEE Access 7:142814–142825. https://doi.org/10.1109/ACCESS.2019.2944755
20. Yu Y, Lei Z, Wang Y, Zhang T, Peng C, Gao S (2022) Improving dendritic neuron model with dynamic scale-free network-based differential evolution. IEEE/CAA J Autom Sin 9(1):99–110. https://doi.org/10.1109/JAS.2021.1004284
21. He H, Gao S, Jin T, Sato S, Zhang X (2021) A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction. Appl Soft Comput 108:107488. https://doi.org/10.1016/j.asoc.2021.107488
22. Todo Y, Tamura H, Yamashita K, Tang Z (2014) Unsupervised learnable neuron model with nonlinear interaction on dendrites. Neural Netw 60:96–103. https://doi.org/10.1016/j.neunet.2014.07.011
23. Yang H, Yu Y, Cheng J, Lei Z, Cai Z, Zhang Z, Gao S (2022) An intelligent metaphor-free spatial information sampling algorithm for balancing exploitation and exploration. Knowl Based Syst. https://doi.org/10.1016/j.knosys.2022.109081
24. Wang Z, Gao S, Zhou M, Sato S, Cheng J, Wang J (2022) Information-theory-based nondominated sorting ant colony optimization for multiobjective feature selection in classification. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2022.3185554
25. Song Z, Tang C, Ji J, Todo Y, Tang Z (2021) A simple dendritic deural network model-based approach $PM_{2.5}$ concentration prediction. Electronics 10(4):373. https://doi.org/10.3390/electronics10040373
26. Partonen T, Haukka J, Nevanlinna H, Lönnqvist J (2004) Analysis of the seasonal pattern in suicide. J Affect Disord 81(2):133–139. https://doi.org/10.1016/S0165-0327(03)00137-X
27. Maciejewski R, Hafen R, Rudolph S, Larew SG, Mitchell MA, Cleveland WS, Ebert DS (2010) Forecasting hotspots-a predictive analytics approach. IEEE Trans Visual Comput Graph 17(4):440–453. https://doi.org/10.1109/TVCG.2010.82
28. Theodosiou M (2011) Forecasting monthly and quarterly time series using STL decomposition. Int J Forecast 27(4):1178–1195. https://doi.org/10.1016/j.ijforecast.2010.11.002
29. Hernández-Santoro C, Contreras-Reyes JE, Landaeta MF (2019) Intra-seasonal variability of sea surface temperature influences phenological decoupling in anchovy (Engraulis ringens). J Sea Res 152:101765. https://doi.org/10.1016/j.seares.2019.101
30. Zhang J, Shao K, Luo X (2018) Small sample image recognition using improved convolutional neural network. J Vis Commun Image Represent 55:640–647. https://doi.org/10.1016/j.jvcir.2018.07.011
31. Yin W, Kann K, Yu M, Schütze H (2017) Comparative study of CNN and RNN for natural language processing. ArXiv:1702.01923
32. Morchid M (2018) Parsimonious memory unit for recurrent neural networks with application to natural language processing. Neurocomputing 314:48–64. https://doi.org/10.1016/j.neucom.2018.05.081
33. Khotanzad A, Elragal H, Lu T-L (2000) Combination of artificial neural-network forecasters for prediction of natural gas consumption. IEEE Trans Neural Netw 11(2):464–473. https://doi.org/10.1109/72.839015
34. Nagy H, Watanabe K, Hirano M (2002) Prediction of sediment load concentration in rivers using artificial neural network model. J Hydraul Eng 128(6):588–595. https://doi.org/10.1061/(ASCE)0733-9429(2002)128:6(588)
35. Wang Y, Yu Y, Cao S, Zhang X, Gao S (2020) A review of applications of artificial intelligent algorithms in wind farms. Artif Intell Rev 53(5):3447–3500. https://doi.org/10.1007/s10462-019-09768-7
36. Chae J, Thom D, Bosch H, Jang Y, Maciejewski R, Ebert DS, Ertl T (2012)Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In: 2012 IEEE conference on visual analytics science and technology (VAST). IEEE, pp 143–152. https://doi.org/10.1109/VAST.2012.6400557
37. Stow CA, Cha Y, Johnson LT, Confesor R, Richards RP (2015) Long-term and seasonal trend decomposition of Maumee River nutrient inputs to western Lake Erie. Environ Sci Technol 49(6):3392–3400. https://doi.org/10.1021/es5062648
38. Chen X, Yin L, Fan Y, Song L, Ji T, Liu Y, Tian J, Zheng W (2020) Temporal evolution characteristics of PM2. 5 concentration based on continuous wavelet transform. Sci Total Environ 699:134244. https://doi.org/10.1016/j.scitotenv.2019.134244
39. Gao S, Wang K, Tao S, Jin T, Dai H, Cheng J (2021) A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models. Energy Convers Manag 230:113784. https://doi.org/10.1016/j.enconman.2020.113784
40. Xu Z, Gao S, Yang H, Lei Z (2021) SCJADE: yet another state-of-the-art differential evolution algorithm. IEEJ Trans Electr Electron Eng 16(4):644–646. https://doi.org/10.1002/tee.23340
41. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713. https://doi.org/10.1109/TEVC.2008.91900
42. Gao S, Zhou M, Wang Y, Cheng J, Yachi H, Wang J (2019) Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. IEEE Trans Neural Netw Learn Syst 30(2):601–614. https://doi.org/10.1109/TNNLS.2018.2846646