A Simple Dendritic Neural Network Model-Based Approach for Daily PM$_{2.5}$ Concentration Prediction

Zhenyu Song $^1$, Cheng Tang $^2$, Junkai Ji $^3$*, Yuki Todo $^4$ and Zheng Tang $^2$

1 College of Computer Science and Technology, Taizhou University, Taizhou 225300, China; songzhenyu@tzu.edu.cn
2 Faculty of Engineering, University of Toyama, Toyama-shi 930-8555, Japan; tangcheng(toyama)gmail.com (C.T.); ztang@eng.u-toyama.ac.jp (Z.T.)
3 College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
4 School of Electrical and Computer Engineering, Kanazawa University, Kanazawa-shi 920-1192, Japan; yktodo@ect.kanazawa-u.ac.jp
* Correspondence: jjunkai@szu.edu.cn

Abstract: Air pollution in cities has a massive impact on human health, and an increase in fine particulate matter (PM$_{2.5}$) concentrations is the main reason for air pollution. Due to the chaotic and intrinsic complexities of PM$_{2.5}$ concentration time series, it is difficult to utilize traditional approaches to extract useful information from these data. Therefore, a neural model with a dendritic mechanism trained via the states of matter search algorithm (SDNN) is employed to conduct daily PM$_{2.5}$ concentration forecasting. Primarily, the time delay and embedding dimensions are calculated via the mutual information-based method and false nearest neighbours approach to train the data, respectively. Then, the phase space reconstruction is performed to map the PM$_{2.5}$ concentration time series into a high-dimensional space based on the obtained time delay and embedding dimensions. Finally, the SDNN is employed to forecast the PM$_{2.5}$ concentration. The effectiveness of this approach is verified through extensive experimental evaluations, which collect six real-world datasets from recent years. To the best of our knowledge, this study is the first attempt to utilize a dendritic neural model to perform real-world air quality forecasting. The extensive experimental results demonstrate that the SDNN offers very competitive performance relative to the latest prediction techniques.

Keywords: air quality forecasting; neural network; environment; PM$_{2.5}$ concentration

1. Introduction

In recent years, with the development of the economy and urban industries, atmospheric pollution has increased and gained global attention. In particular, air pollution is increasingly serious and threatens our living environments and human health. In the air, high concentrations of fine particulate matter (PM$_{2.5}$ and fine aerosols with a particle size of less than or equal to 2.5 µm) are the main pollutants [1]. The composition of fine particles is very complex and difficult to control and contains various hazardous and toxic substances. To protect the environment and human health, many countries are incorporating environmental governance into their development strategies, and many observation stations have been built to monitor real-time PM$_{2.5}$ concentrations. Based on reliable and accurate forecasting values, announcing the concentration of pollutants days or hours in advance can help the public become aware of this hazard and make early-warning decisions. Therefore, PM$_{2.5}$ concentration prediction is very important for environmental management.

The precise forecasting of PM$_{2.5}$ concentrations is a challenging task due to their diverse impacts, irregular properties and chaotic nonlinear characteristics. As one of the most crucial methods for assessing air quality, PM$_{2.5}$ forecasting has become a major research focus in air pollution research. In addition, some researchers have begun to predict and
analyse other specific pollutant concentrations, such as NO\textsubscript{2} [2,3], PM\textsubscript{10} [2,4], SO\textsubscript{2} [2], and the air quality index [5]. In general, the methods proposed for PM\textsubscript{2.5} forecasting can be mainly divided into deterministic approaches, statistical approaches and machine learning methods. Deterministic approaches are typically knowledge-based approaches that use chemical and physical theories to simulate the transformation and transportation of air pollutants for forecasting. However, relevant studies have verified that deterministic approaches have difficulty accurately predicting PM\textsubscript{2.5} concentrations since they cannot be used to describe the nonlinear relationships and time-varying characteristics of data [6].

In contrast, statistical approaches generally apply data and use regression methods and time series theory to explain the correlation between historical and future data. These methods are also considered simpler and more efficient than knowledge-based deterministic methods [7].

Because of the irregularity and non-linearity of PM\textsubscript{2.5} concentration data, these statistical methods cannot obtain more reliable and accurate prediction results to satisfy the requirements of practical application. To overcome this limitation, various machine learning methods have been recently proposed for PM\textsubscript{2.5} concentration prediction, such as the random forest [8] and support vector regression (SVR) [9] methods. In addition, due to their assorted memory, self-learning, data-adaptable, and data-driven characteristics, many researchers pay attention to artificial neural networks (ANNs), which can learn to accurately and reliably map the correlations between inputs and outputs. However, it is difficult to select the most suitable ANN for different PM\textsubscript{2.5} concentration time series because each one has its own advantages and limitations. Accordingly, considering the calculation costs and feasibility of the method, we attempt to improve the PM\textsubscript{2.5} prediction performance using a very simple ANN named the dendritic neural network model (DNN), which was proposed in our previous studies [10]. The DNN uses a multiplicative operation to capture the nonlinear relationships between features. Compared to other ANNs, the DNN can be considered a more realistic neuron model, since it considers the nonlinear computation of synapses and dendritic structures, which is inspired by the biological phenomena in neurons [11]. Such models have been successfully employed for various applications such as computer-aided medical diagnosis [12], time series prediction [13], and morphological hardware realization [14]. However, the original DNN and simplified variation with a single branch (S-DNN) are trained by an error back-propagation (BP) algorithm. The BP algorithm is based on gradient descent information, which makes it easily fall into local optima and thus sensitive to initial conditions, overfitting and slow convergence. These disadvantages largely limit the performance of the DNN and its variations. To overcome these issues, it is necessary to identify a more powerful learning algorithm to train the DNN. In this paper, a recently proposed heuristic optimization algorithm, which is named the states of matter search (SMS) algorithm [15], is selected to optimize the weights and thresholds of the DNN and utilize it for PM\textsubscript{2.5} concentration time series prediction. The evolutionary processes of the SMS can be divided into a gas state, a liquid state, and a solid state. In each state, the positions of the agents are updated based on the direction vector operator, collision operator, and random behaviour. As a global search algorithm, the SMS algorithm offers powerful optimization abilities that can effectively avoid local optima during the training phase and significantly enhance the prediction accuracy of the DNN.

Since real-world PM\textsubscript{2.5} concentration time series are based on one-dimensional, irregular and unpredictable data and should be mapped to a high-dimensional space based on a certain time delay and embedding dimension, some intrinsic properties will be revealed. Takens’ theorem is a commonly used approach [16] that applies the phase space reconstruction (PSR) approach to transform these time series data into new high-dimensional embedding spaces while preserving the topological structure of the chaotic attractors. Therefore, we calculate the time delay using the mutual information (MI)-based method [17], and embedding dimensions are obtained by the false nearest-neighbour (FNN) approach [18]. Then, the PSR is performed depending on the time delay and embedding dimensions, and the maximum Lyapunov exponent (MLE) is used to detect the predictability and chaotic
properties [19]. Finally, the trained SDNN is used to forecast the PM$_{2.5}$ concentration. In our experiments, six PM$_{2.5}$ concentration datasets are used to evaluate the prediction performance of the SDNN. The SMS training results are compared to those of seven other optimization algorithms, and the prediction performance of the SDNN is compared to the results of some competitive forecasting approaches. To obtain reliable results, each experiment is independently performed 30 times. The experimental and statistical analysis results suggest that the SDNN can achieve very competitive prediction results. Moreover, in order to verify whether the proposed method can be applied to more time series predictions, we discuss the simulations on an open available PM$_{2.5}$ dataset from UCI machine learning repository.

The main contributions of this study are as follows: (1) A more realistic SDNN that considers nonlinear computation in dendritic structures and synapses is applied to PM$_{2.5}$ concentration prediction for the first time. (2) To enhance the prediction stability and accuracy, a global optimization algorithm named the SMS is selected to train the SDNN. Experimental results show that compared to other state-of-the-art prediction approaches, the DNN obtains prominent competitive performance for PM$_{2.5}$ concentration forecasting. (3) The study shows that expanding the application scope of the DNN for prediction problems can help us better understand the capacities of the DNN.

The remainder of this paper is organized as follows. Section 2 introduces some related works on PM$_{2.5}$ concentration forecasting. Section 3 elaborates on the SDNN, SMS algorithm and relevant methods to predict the PM$_{2.5}$ concentration time series in detail. Sections 4 and 5 present our parameter settings, experimental and statistical results and a discussion, respectively. The Section 6 draws conclusions.

2. Related Work

In the literature, various ANN architectures have provided strong advantages in PM$_{2.5}$ concentration forecasting, such as back-propagation (BP) neural networks [20], fuzzy neural networks [21] and long short-term memory (LSTM) neural networks [22]. Specifically, Xu and Yoneda employed the LSTM auto-encoder multi-task learning model for air quality prediction in [23]. The employment of a recurrent neural network (RNN) to forecast the air quality is presented in [24,25], and more RNN architectures for multi-sequence indoor PM$_{2.5}$ concentration prediction are compared and analysed in [25]. In [21], Lin et al. proposed a neuron-fuzzy modelling system for forecasting. In addition, several deep learning models have been successfully applied in air quality forecasting [26–28]. More references regarding the ANN-based PM$_{2.5}$ concentration prediction approaches can be found in [29–35].

In addition to the above methods, hybrid models are another popular choice for air quality prediction in the literature. Feng et al. proposed a hybrid model that combined a geographic model, wavelet transformation analysis and ANN to enhance air quality forecasting accuracy [36]. The combination of the ANN and multiple linear and continuous regression models are introduced in [37]. Sun et al. developed a novel approach based on the least-square SVM and principal component analysis technique [38], and an integrated model composed of SVM and autoregressive integrated moving average model is presented in [5]. Liu et al. utilized a multi-resolution multi-objective ensemble model for PM$_{2.5}$ prediction [39]. Qi et al. integrated the LSTM and graph convolutional networks to model PM$_{2.5}$ forecasting [40]. Combined with feature extraction based on the ensemble empirical mode decomposition approach, Bai et al. applied the LSTM approach to PM$_{2.5}$ concentration prediction [22]. The hybrid model based on a BP neural network and convolutional neural network can make accurate PM$_{2.5}$ predictions in [41]. A hybrid prediction model using land use regression and a chemical transport model can be found in [42]. Overall, although various machine learning techniques and hybrid methods are widely applied for air quality forecasting, which can achieve satisfactory prediction performance to a certain degree, they consume large amounts of calculation costs.
3. Methodology Formulation

3.1. SDNN Structure

The original SDNN is inspired by the dendritic mechanism of biological neurons. It is composed of three layers: a synaptic layer, a dendritic layer and a soma layer. The weights and thresholds are trained by the optimization algorithm. The structural morphology of the SDNN is shown in Figure 1, which has \( M \) dendritic branches and \( n \) synaptic layers depending on specific problems, and \( a_1 \sim a_n \) are the attributes of a certain problem. Incoming signals \( a_1 \sim a_n \) from the synaptic layer enter the dendritic structure through synapses. Then, the results for each dendritic layer are collected and sent to the soma layer. A mathematical description of the SDNN is provided as follows.

![Figure 1. Architectural description of the SDNN.](image)

3.1.1. Synapses

The synaptic layer is the synaptic connection structure from the dendrite of a neuron, and each synapse receives the incoming signal from the feature attributes of the training data and transfers it to the next layer through a sigmoid function. The computation operator that describes the \( j \)-th (\( j = 1, 2, \ldots, M \)) branch receiving the \( i \)-th (\( i = 1, 2, \ldots, n \)) input is expressed as follows:

\[
S_{ij} = \left( 1 + e^{-K(w_{ij} \times sa_i - q_{ij})} \right)^{-1},
\]

where \( S_{ij} \) is the result of the \( i \)-th synapse for the \( j \)-th dendritic branch and \( K \) is a positive constant. Synaptic parameters \( w_{ij} \) and \( q_{ij} \) must be trained by the training algorithm. According to \( q_{ij} \) and \( w_{ij} \), the synaptic layers have four connection cases, which are illustrated in Figure 2. Moreover, threshold \( \alpha_{ij} \) for the synaptic layer is obtained from \( \alpha_{ij} = q_{ij}/w_{ij} \).

- Case (Constant-1 connection): When \( q_{ij} < w_{ij} < 0 \) or \( q_{ij} < 0 < w_{ij} \), in this case, the output of the synapse is always approximately 1 despite the changes in the input.
- Case (Constant-0 connection): When \( 0 < w_{ij} < q_{ij} \) or \( w_{ij} < 0 < q_{ij} \), in this case, the result is always 0 despite the changes in the input.
- Case (Inverse connection): When \( w_{ij} < q_{ij} < 0 \), where \( a_i > \alpha_{ij} \), the output is approximately 0; otherwise, the output tends to be 1.
- Case (Direct connection): When \( 0 < q_{ij} < w_{ij} \), where \( a_i > \alpha_{ij} \), the output tends to be 1; otherwise, the output is approximately 0.
3.1.2. Dendrites

This layer performs a nonlinear operation for the incoming signals of each dendritic branch. The simplest multiplication operation plays a significant role in the processing and transmission of neural computation [43], which is calculated by the following equation:

$$D_j = \prod_{i=1}^{n} S_{ij},$$  \hspace{1cm} (2)

3.1.3. Soma

The soma is the core part of the neuron. First, the soma layer accumulates signals from all dendritic branches and performs the summation function from the previous layer. Then, the results are transferred to the soma, where a sigmoid function is commonly employed to represent the computational process of this layer. The soma can be described by the following equation:

$$\text{Soma} = \left(1 + e^{-K_s(\sum_{j=1}^{M} D_j - \beta)}\right)^{-1},$$ \hspace{1cm} (3)

where $\beta$ is a user-defined constant threshold, $K_s$ is an adjustable constant parameter, and Soma is the final output of the model.

3.2. Training Algorithm

The multiplication operation is applied to each dendritic branch of the SDNN, which makes the results of the SDNN extremely sensitive to each attribute. Moreover, the parameter space of the SDNN is very complex and large. Thus, this situation requires an optimization algorithm, which has powerful search ability for the SDNN optimization. In this study, a swarm-based optimization algorithm, which is called the SMS algorithm, is adopted as a training algorithm to optimize the parameters of the SDNN. In this section, the SMS algorithm is briefly described in more detail.

The SMS algorithm emulates the states of matter phenomenon [38], and a population of optimized agents is described as molecules that interact with one another by evolutionary operators based on the physical principles of the thermal-energy motion ratio. The evolutionary process of the SMS algorithm can be divided into three phases: (1) a gas state, (2) a liquid state, and (3) a solid state. The agents have different exploitation-exploration energies in each stage. In the first (gas) state, agents experience severe collisions...
and motions at the beginning of the optimization process. The second state is the liquid state, which restricts the collision and movement energy of agents more than the gas state. The final state is the solid state, where individuals are prevented from freely moving due to the forces among them. The overall optimization process of the SMS algorithm is described in Figure 3.

![Evolution process](image)

**Figure 3.** Evolutionary process of the SMS algorithm.

In this optimization algorithm, the agents are considered molecules whose positions change when the process iterates. The movement of these molecules is analogous to the motion governing heat, which depends on three optimization states: (1) direction vector operator, (2) collision operator, and (3) random behaviour.

### 3.2.1. Direction Vector

First, the SMS algorithm randomly generates a position for each agent. Position $P_i$ of each agent is described as vector $d_i$ in the search space. When the process evolves, the direction vector operator provides an attraction phenomenon by moving each molecule towards the current best particle. Thus, these direction vectors are iteratively updated and can be defined as follows:

$$
d_{i+1} = d_i \times \left( 1 - \frac{t}{i_{\text{max}}} \right) \times 0.5 + \frac{p_{\text{best}} - P_i}{\|p_{\text{best}} - P_i\|},$$

where $p_{\text{best}}$ is the current best individual seen thus far and $t$ and $i_{\text{max}}$ are the current iteration number and maximum number of iterations, respectively. Once the direction has been determined, we can calculate the velocity vector as follows:

$$
v_i = d_i \times \frac{\sum_{m=1}^{n}(b_{\text{high}}^m - b_{\text{low}}^m)}{n} \times \gamma,$$

where $b_{\text{high}}^m$ and $b_{\text{low}}^m$ are the upper and lower $m$-th parameter bounds, respectively, and $\gamma \in [0, 1]$. $n$ is the number of decision variables. Once the direction and velocity are obtained from these two equations, the new position of each molecule is calculated from:

$$
p_{i+1}^{t+1} = p_{i,m} + v_m \times \text{rand}(0, 1) \times \left( b_{\text{high}}^m - b_{\text{low}}^m \right) \times \alpha,$$

where $\alpha \in [0.5, 1]$ and rand $(0, 1)$ is a random number between 0 and 1.

### 3.2.2. Collisions

The collision operator emulates the collision phenomenon, where molecules interact with one another if the distances among these molecules are shorter than a proximity collision radius, and the collision operator provides a diversity of individuals, which prevents premature convergence. The collision radius is defined as follows:

$$
r = \frac{\sum_{m=1}^{n}(b_{\text{high}}^m - b_{\text{low}}^m)}{n} \times \beta,$$
where $\beta \in [0, 1]$. If two molecules ($P_i$ and $P_m$) have collided, the direction vectors of the two molecules ($d_i$ and $d_m$) are modified by exchanging their direction vectors as follows:

$$d_i = d_m \quad \text{and} \quad d_m = d_i.$$  

### 3.2.3. Random Behaviour

The transition of molecules from one state to another commonly exhibits random behaviour. The SMS algorithm allows molecules to randomly change position by following a probabilistic criterion in a feasible space, which can be defined as follows:

$$p_{i,m}^{t+1} = \begin{cases} b_{m}^{\text{low}} + \text{rand}(0, 1) \times (b_{m}^{\text{high}} - b_{m}^{\text{low}}) & \text{with probability } H \\ p_{i,m}^{t+1} & \text{with probability } (1 - H) \end{cases},$$  

where $H$ is a probability depending on the current SMS state. Furthermore, $m \in \{1, ..., n\}$. Based on different states, the SMS algorithm controls the motion operator by adjusting parameters $\gamma$, $\alpha$, $\beta$, and $H$. The values of these parameters are provided by [38] and summarized in Table 1.

### Table 1. Parameter settings of the SMS algorithm.

| State  | Duration | $\gamma$ | $\alpha$ | $\beta$ | Probability $H$ |
|--------|----------|----------|----------|---------|----------------|
| Gas    | 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, 0.1] | 0.0           |

### 3.3. Time Delay and Embedding Dimensions

According to chaos theory, a PM$_{2.5}$ concentration time series can be mapped into a high-dimensional space by the PSR. To perform PSR, time delay $\tau$ and embedding dimensions $m$ are necessary, which can be calculated by the MI approach and FNN method, respectively. Then, the MLEs are calculated to detect the chaos characteristics of the PM$_{2.5}$ concentration data.

The MI approach has gained broad acceptance as a metric of association between variables, which measures both nonlinear and linear correlations. Parameter time delay $\tau$ is employed to map one-dimensional data to a higher-dimensional space, where each point is independent and identically distributed. A suitable time delay value can ensure that the data points are highly correlated, independent, smooth, and identifiable [44]. According to the information entropy theory, $\tau$ can be determined from the MI $I(a_t, a_{t+\tau})$, which is described as follows:

$$I(\tau) = \sum_{t=1}^{N-\tau} P(a_t, a_{t+\tau}) \log_2 \left( \frac{P(a_t, a_{t+\tau})}{P(a_t)P(a_{t+\tau})} \right),$$  

where $P(a_t, a_{t+\tau})$ is the joint probability; $P(a_t)$ and $P(a_{t+\tau})$ are the marginal probabilities of $a_t$ and $a_{t+\tau}$, respectively. The optimal $\tau$ is a possible integer and determined by the first minimum value of $I(\tau)$.

In addition, the FNN method is utilized to calculate the embedding dimensions. Similarly, an appropriate embedded dimension value must ensure the behaviour of the original data and maintain relevance among the data. The optimal $m$ value is also a positive integer and can be obtained by the first minimum value of the FNN rate. This method employs two conditions to evaluate the points as false neighbours, which are described as follows:

- Calculate Euclidean distance $D_1$ between point $a_i$ and its nearest point $a_i^{NN}$. Both $a_i$ and $a_i^{NN}$ are joined by the dimension from $d$ to $d + 1$; then, compute the new Euclidean
distance $D_2$. If the result is greater than threshold $\mu$, then the points are considered false neighbours; otherwise, verify the second condition.

$$ \left( \frac{D_2 - D_1}{D_1} \right)^{1/2} = \frac{|a_{1+Td_2} - a'_{1+Td_2}|}{D_1} \geq \mu $$

(11)

- If $D_2$ cannot satisfy the following condition, then the points are considered false neighbours.

$$ \frac{D_2}{\delta_{pm}} \geq V_{tol} $$

(12)

where $\delta_{pm}$ is the standard deviation of the PM$_{2.5}$ concentration time series, and $V_{tol}$ is the positive integer threshold that describes the attractor size.

3.4. PSR and the MLE

According to long-term monitoring, the real-world PM$_{2.5}$ concentration time series shows chaoticity and unpredictability. Hence, it is quite difficult to make accurate predictions. Nevertheless, the periodicity is proved when reconstructed as points of the phase space. The PSR technique applies a basic theory of chaotic dynamic systems widely utilized in the analysis of nonlinear systems [39]. It is confirmed that PSR can expand the time series into a new space while preserving the topological structure of the high-dimensional space with chaotic attractors. Crucial factors $\tau$ and $m$ of PSR for the real-world PM$_{2.5}$ concentration time series can be obtained using the above methods. Thus, PSR and target data $T$ can be expressed as follows:

$$ P = \begin{bmatrix} a_1 & a_2 & \cdots & a_{N-(m-1)\tau-1} \\ a_{1+\tau} & a_{2+\tau} & \cdots & a_{N-(m-2)\tau-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1+(m-1)\tau} & a_{2+(m-1)\tau} & \cdots & a_{N-1} \end{bmatrix}, \quad T = \begin{bmatrix} a_{2+(m-1)\tau} \\ a_{3+(m-1)\tau} \\ \vdots \\ a_N \end{bmatrix} $$

(13)

The MLE is generally employed to confirm the properties of chaotic dynamics [45] and estimate whether a sequence has chaotic characteristics. In general, the sequence motion is chaotic only if the value of the MLE becomes positive [45]. In this study, we use this approach to identify the chaotic characteristics of the PM$_{2.5}$ concentration time series. The MLE can be calculated as follows:

$$ \text{MLE} = \frac{1}{t_M - t_0} \sum_{i=0}^{M} L_i' \ln \frac{L_i'}{L_i} $$

(14)

where $t = 1, 2, \ldots, N - (m - 1)\tau$. We assume that a phase point and the initial time are $t_0$ and $a(t_0)$, respectively. $L_0$ is the minimum distance from a neighbouring phase point. In addition, we set the distance $L_0'(|a(t_1) - a(t_0)|)$ to be larger than a positive threshold at time $t_1$. $L_0'$ is replaced with $L_1'$ when the next distance $L_1$ to another phase point is greater than $L_0$ at time $t_2$, and this computational process continues until the last phase point $a_N$.

As mentioned above, a dynamic system manifests chaotic characteristics when the MLE exceeds 0, and the value of the MLE is typically 0–1 to enable the long-term prediction [46].

Figure 4 shows the time delays and embedding dimensions of six PM$_{2.5}$ concentration time series based on the MI approach and FNN method, respectively. The results of the time delay and embedding dimensions are obtained for each training dataset, and the computational results of $\tau$, $m$ and MLE for all PM$_{2.5}$ concentration datasets are summarized in Table 2.
Figure 4. Relationship between the mutual information (MI) and time delay and between the false nearest-neighbour (FNN) rate and the embedding dimensions for a PM$_{2.5}$ concentration time series.

Table 2. Resulting time delay, embedding dimensions and maximum Lyapunov exponent (MLE) of the PM$_{2.5}$ concentration time series for phase space reconstruction (PSR).

| Dataset       | Time Delay $\tau$ | Embedding Dimension $m$ | MLE     |
|---------------|-------------------|-------------------------|---------|
| PM$_{2.5}$ data 1 | 3                 | 4                       | 0.0528  |
| PM$_{2.5}$ data 2 | 5                 | 4                       | 0.0023  |
| PM$_{2.5}$ data 3 | 5                 | 4                       | 0.0024  |
| PM$_{2.5}$ data 4 | 4                 | 5                       | 0.0380  |
| PM$_{2.5}$ data 5 | 4                 | 4                       | 0.0474  |
| PM$_{2.5}$ data 6 | 4                 | 4                       | 0.0452  |

4. Experiments

In this study, all prediction models are evaluated on six PM$_{2.5}$ concentration datasets obtained from the Beijing Monitoring Center Station in China in eastern Asia. Predictions are made 1 day ahead.

4.1. Dataset Description

Our experiment uses six real-world daily PM$_{2.5}$ concentration datasets collected from the Ministry of Ecology and Environment of China over 4 years and 6 months (1 January 2016 to 30 June 2020). We select six datasets of 2-year terms and divide them into two subsets: a training set and a prediction set. The number of instances used for training and prediction are approximately the top 75% and bottom 25%, respectively, of each set of data. The details of the experimental datasets are presented in Table 3.

Table 3. Description of the Beijing daily PM$_{2.5}$ concentration experimental datasets.

| Dataset      | Training Interval Year/Month | Prediction Interval Year/Month | Instance Number Days |
|--------------|-----------------------------|--------------------------------|----------------------|
| PM$_{2.5}$ data 1 | 2016/01–2017/06             | 2017/07–2017/12                | 547,184              |
| PM$_{2.5}$ data 2 | 2016/07–2017/12             | 2018/01–2018/06                | 549,181              |
| PM$_{2.5}$ data 3 | 2017/01–2018/06             | 2018/07–2018/12                | 546,184              |
| PM$_{2.5}$ data 4 | 2017/07–2018/12             | 2019/01–2019/06                | 549,181              |
| PM$_{2.5}$ data 5 | 2018/01–2019/06             | 2019/07–2019/12                | 546,184              |
| PM$_{2.5}$ data 6 | 2018/07–2019/12             | 2020/01–2020/06                | 549,182              |

4.2. Normalization

First, to improve the computation speed and reduce the computation complexity, we normalize all inputs to a range of [0, 1] based on the following equation:

$$a'_i = \frac{a_i - a_{\min}}{a_{\max} - a_{\min}}$$  (15)
where \( a_{\text{min}} \) and \( a_{\text{max}} \) are the minimal and maximal values, respectively, of the original vector. Notably, normalization is performed during both the training phase and testing phase. In addition, the inverse normalization operation is performed on the outputs of the model.

4.3. Parameter Settings

As mentioned above, 3 hyperparameters impact the performance of the SDNN: \( K \), \( M \), and \( \beta \). \( K \) is a positive integer value of the sigmoid function in the synaptic layer of the SDNN; \( M \) is the number of branches in the dendritic layer, which is commonly greater than the feature number; \( \beta \) is a threshold of the soma layer.

In general, the exhaustive approach employed to determine these parameters is resource intensive. To achieve the best performance while simultaneously decreasing the material, labour, and time costs, Taguchi’s method, which utilizes orthogonal arrays to find a reasonable parameter combination for each dataset, is used to reduce the number of experimental runs [47]. Then, \( L_{16}(4^3) \) orthogonal arrays are generated, which cover only 16 (of 64) experiments in the preliminary work. To achieve reliable average performance, we perform each experiment over 30 independent runs using Taguchi’s method, and the experimental results are summarized in Table 4. Each dataset clearly corresponds to a set of optimal parameter combinations. In addition, the population size and maximum number of iterations are set to 50 and 1000, respectively.

4.4. Evaluation Criteria

To perform a comprehensive performance comparison, the performance of each approach can be assessed by five commonly utilized metrics: the mean squared error of the predictions (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error of the predictions (RMSE), and correlation exponents of the prediction (CEs), which are defined by the following formulas:

- The MSE of the predictor for the normalized data is obtained as follows:
  \[
  \text{MSE}(f, \hat{f}) = \frac{1}{n} \sum_{i=1}^{n} (\hat{f}_i - f_i)^2. \tag{16}
  \]

- The MAE is defined as follows:
  \[
  \text{MAE}(f, \hat{f}) = \frac{1}{n} \sum_{i=1}^{n} |f_i - \hat{f}_i|. \tag{17}
  \]

- The MAPE of the predictions is defined as follows:
  \[
  \text{MAPE}(f, \hat{f}) = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{f_i - \hat{f}_i}{f_i} \right|. \tag{18}
  \]

- The RMSE for the normalized distribution can be defined as follows:
  \[
  \text{RMSE}(f, \hat{f}) = \left[ \frac{1}{n} \sum_{i=1}^{n} (|f_i - \hat{f}_i|)^2 \right]^{1/2}. \tag{19}
  \]

- The CEs of the prediction phase can be given by the following:
  \[
  \text{CE}(f, \hat{f}) = \left[ \frac{\sum_{i=1}^{n} (\hat{f}_i - \bar{\hat{f}})(f_i - \bar{f})}{\left( \sum_{i=1}^{n} (\hat{f}_i - \bar{\hat{f}})^2 \right)^{1/2}\left( \sum_{i=1}^{n} (f_i - \bar{f})^2 \right)^{1/2}} \right]. \tag{20}
  \]

where \( \hat{f}_i \) is a target vector, \( f_i \) is the output of the utilized prediction model, and \( n \) is the number of instances.
Table 4. Experimental results of the SDNN for the PM$_{2.5}$ concentration data and different parameter combinations.

| Parm. | PM$_{2.5}$ Concentration Dataset |
|-------|----------------------------------|
| NO.   | K    | M    | β    | Data 1 (Mean ± Std) | Data 2 (Mean ± Std) | Data 3 (Mean ± Std) | Data 4 (Mean ± Std) | Data 5 (Mean ± Std) | Data 6 (Mean ± Std) |
| 1     | 6    | 4    | 0.3  | 7.60 × 10^{-3} ± 1.06 × 10^{-2} ± 1.04 × 10^{-2} ± 1.78 × 10^{-2} ± 1.28 × 10^{-2} ± 1.92 × 10^{-2} ± |
| 2     | 6    | 7    | 0.6  | 5.26 × 10^{-3} ± 7.67 × 10^{-3} ± 7.87 × 10^{-3} ± 1.67 × 10^{-2} ± 1.19 × 10^{-2} ± 1.83 × 10^{-2} ± |
| 3     | 6    | 10   | 0.9  | 5.26 × 10^{-3} ± 7.66 × 10^{-3} ± 7.81 × 10^{-3} ± 1.72 × 10^{-2} ± 1.19 × 10^{-2} ± 1.82 × 10^{-2} ± |
| 4     | 6    | 13   | 1.2  | 5.21 × 10^{-3} ± 7.78 × 10^{-3} ± 7.71 × 10^{-3} ± 1.68 × 10^{-2} ± 1.19 × 10^{-2} ± 1.89 × 10^{-2} ± |
|      | 5    | 10   | 0.6  | 5.13 × 10^{-3} ± 7.73 × 10^{-3} ± 8.01 × 10^{-3} ± 1.68 × 10^{-2} ± 1.17 × 10^{-2} ± 1.84 × 10^{-2} ± |
|      | 6    | 10   | 0.3  | 5.22 × 10^{-3} ± 7.59 × 10^{-3} ± 7.95 × 10^{-3} ± 1.72 × 10^{-2} ± 1.19 × 10^{-2} ± 1.88 × 10^{-2} ± |
|      | 7    | 10   | 1.2  | 5.19 × 10^{-3} ± 7.69 × 10^{-3} ± 7.98 × 10^{-3} ± 1.72 × 10^{-2} ± 1.18 × 10^{-2} ± 1.91 × 10^{-2} ± |
|      | 8    | 10   | 0.9  | 5.20 × 10^{-3} ± 7.65 × 10^{-3} ± 7.91 × 10^{-3} ± 1.74 × 10^{-2} ± 1.19 × 10^{-2} ± 1.97 × 10^{-2} ± |
|      | 9    | 14   | 0.9  | 5.06 × 10^{-3} ± 7.53 × 10^{-3} ± 7.99 × 10^{-3} ± 1.67 × 10^{-2} ± 1.17 × 10^{-2} ± 1.87 × 10^{-2} ± |
|      | 10   | 14   | 1.2  | 5.15 × 10^{-3} ± 7.68 × 10^{-3} ± 7.93 × 10^{-3} ± 1.74 × 10^{-2} ± 1.17 × 10^{-2} ± 1.93 × 10^{-2} ± |
|      | 11   | 14   | 0.3  | 5.25 × 10^{-3} ± 7.62 × 10^{-3} ± 8.18 × 10^{-3} ± 1.78 × 10^{-2} ± 1.19 × 10^{-2} ± 1.96 × 10^{-2} ± |
|      | 12   | 14   | 0.6  | 5.20 × 10^{-3} ± 7.56 × 10^{-3} ± 8.14 × 10^{-3} ± 1.79 × 10^{-2} ± 1.19 × 10^{-2} ± 2.02 × 10^{-2} ± |
|      | 13   | 18   | 1.2  | 5.05 × 10^{-3} ± 7.97 × 10^{-3} ± 8.00 × 10^{-3} ± 1.67 × 10^{-2} ± 1.16 × 10^{-2} ± 1.88 × 10^{-2} ± |
|      | 14   | 18   | 0.9  | 5.10 × 10^{-3} ± 7.54 × 10^{-3} ± 8.10 × 10^{-3} ± 1.73 × 10^{-2} ± 1.18 × 10^{-2} ± 1.97 × 10^{-2} ± |
|      | 15   | 18   | 0.6  | 5.14 × 10^{-3} ± 8.13 × 10^{-3} ± 8.34 × 10^{-3} ± 1.74 × 10^{-2} ± 1.17 × 10^{-2} ± 2.03 × 10^{-2} ± |
|      | 16   | 18   | 0.3  | 5.23 × 10^{-3} ± 7.81 × 10^{-3} ± 8.27 × 10^{-3} ± 1.78 × 10^{-2} ± 1.19 × 10^{-2} ± 2.04 × 10^{-2} ± |
4.5. Performance Comparison

In our study, six optimization algorithms and nine prediction models are utilized as competitors of the SDNN. To achieve a reliable evaluation, the experiments for each approach and model are independently repeated 30 times. All experiments are performed on a PC equipped with a 3.80 GHz Intel(R) Core(TM) i7-10770k CPU and 32 GB of RAM using MATLAB R2018b.

4.5.1. Comparison with Other Optimization Algorithms

In this section, we compare the training performance of the SMS algorithm to the performance of seven other optimization algorithms: genetic algorithm (GA) [48], cuckoo search (CS) [49], firefly algorithm (FA) [50], gravitational search algorithm (GSA) [51], adaptive differential evolution with an optional external archive (JADE) [52], adaptive differential evolution with linear population size reduction (L-SHADE) algorithm [53], and particle swarm optimization (PSO) [54]. To ensure the performance of these algorithms, the initial hyperparameters are obtained from the literature listed above. The maximum particle swarm optimization (PSO) [54]. To ensure the performance of these algorithms, the initial hyperparameters are obtained from the literature listed above. The maximum number of iterations is 1000, and the population size is set to 50 for all optimization algorithms. The optimization algorithms are separately employed to train the DNN for PM$_{2.5}$ concentration forecasting. For a fair comparison, we select the identical parameter combination for the SDNN for each dataset. The results achieved by these optimization algorithms for six prediction problems over 30 runs are summarized in Table 5.

Table 5. Experimental results of the dendritic neural network model (DNN) for the PM$_{2.5}$ concentration data and different training optimization algorithms.

| Algorithm | PM$_{2.5}$ Data 1 | PM$_{2.5}$ Data 2 | PM$_{2.5}$ Data 3 | PM$_{2.5}$ Data 4 | PM$_{2.5}$ Data 5 | PM$_{2.5}$ Data 6 |
|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| GA        | 6.27 x 10$^{-3}$ ± 1.47 x 10$^{-3}$ | 1.03 x 10$^{-2}$ ± 2.36 x 10$^{-2}$ | 8.37 x 10$^{-3}$ ± 3.11 x 10$^{-4}$ | 1.91 x 10$^{-2}$ ± 2.58 x 10$^{-4}$ | 1.35 x 10$^{-2}$ ± 4.02 x 10$^{-3}$ |
| CS        | 6.05 x 10$^{-3}$ ± 2.07 x 10$^{-3}$ | 1.08 x 10$^{-2}$ ± 2.63 x 10$^{-3}$ | 8.48 x 10$^{-3}$ ± 3.10 x 10$^{-4}$ | 1.85 x 10$^{-2}$ ± 4.88 x 10$^{-4}$ | 1.27 x 10$^{-1}$ ± 1.24 x 10$^{-2}$ |
| FA        | 5.97 x 10$^{-3}$ ± 1.14 x 10$^{-3}$ | 1.09 x 10$^{-2}$ ± 3.64 x 10$^{-3}$ | 8.48 x 10$^{-3}$ ± 3.79 x 10$^{-4}$ | 1.88 x 10$^{-2}$ ± 2.87 x 10$^{-4}$ | 1.41 x 10$^{-2}$ ± 2.39 x 10$^{-2}$ |
| GSA       | 1.75 x 10$^{-3}$ ± 2.62 x 10$^{-3}$ | 2.46 x 10$^{-2}$ ± 5.55 x 10$^{-3}$ | 1.27 x 10$^{-2}$ ± 5.84 x 10$^{-3}$ | 1.21 x 10$^{-2}$ ± 1.21 x 10$^{-2}$ | 1.29 x 10$^{-2}$ ± 1.96 x 10$^{-2}$ |
| JADE      | 5.27 x 10$^{-3}$ ± 5.25 x 10$^{-4}$ | 9.90 x 10$^{-3}$ ± 3.15 x 10$^{-3}$ | 8.54 x 10$^{-3}$ ± 4.18 x 10$^{-4}$ | 1.80 x 10$^{-2}$ ± 1.49 x 10$^{-3}$ | 1.29 x 10$^{-2}$ ± 2.52 x 10$^{-3}$ |
| L-SHADE   | 4.99 x 10$^{-3}$ ± 3.18 x 10$^{-4}$ | 8.41 x 10$^{-3}$ ± 7.92 x 10$^{-3}$ | 1.51 x 10$^{-2}$ ± 1.51 x 10$^{-2}$ | 1.22 x 10$^{-2}$ ± 1.22 x 10$^{-2}$ | 1.87 x 10$^{-2}$ ± 9.21 x 10$^{-4}$ |
| PSO       | 5.10 x 10$^{-4}$ ± 1.66 x 10$^{-4}$ | 8.62 x 10$^{-3}$ ± 6.20 x 10$^{-4}$ | 8.53 x 10$^{-3}$ ± 1.30 x 10$^{-3}$ | 1.23 x 10$^{-2}$ ± 4.21 x 10$^{-3}$ | 1.88 x 10$^{-2}$ ± 7.17 x 10$^{-3}$ |
| SMS       | 5.05 x 10$^{-5}$ ± 6.37 x 10$^{-5}$ | 7.53 x 10$^{-3}$ ± 7.71 x 10$^{-3}$ | 7.71 x 10$^{-3}$ ± 1.67 x 10$^{-2}$ | 7.60 x 10$^{-3}$ ± 7.60 x 10$^{-3}$ | 1.73 x 10$^{-2}$ ± 8.60 x 10$^{-4}$ |

The SMS algorithm achieves smaller mean and lower standard deviation of MSE for most PM$_{2.5}$ concentration datasets, which implies that the SMS algorithm has more powerful optimization capabilities than the other methods. The exception is that L-SHADE provides the best performance on one of the datasets due to its powerful search ability. To further demonstrate the effectiveness of the SMS algorithm, a nonparametric statistical method called Friedman’s test is used to detect significant differences among multiple groups. Friedman’s test provides a list of ranks to evaluate the performance of all schemes. A lower rank indicates better performance. The average ranks of the seven optimization algorithms for the six PM$_{2.5}$ concentration data prediction problems are listed in Table 6, which shows that the SMS achieves the best performance (ranked 1st), while L-SHADE is the second-best method. Moreover, based on the unadjusted p-values (the probability of several false discoveries), the family-wise error rate is typically ignored for multiple pairwise comparisons. In general, a post hoc test approach called the Bonferroni-Dunn procedure is used to adjust the p-values, which are defined as the $p_{bonf}$ value. The corresponding significance level is set to 0.1. Through the above method, the $p_{bonf}$ values are calculated and presented in Table 6. These statistical results imply that the SMS algorithm is significantly better than the GA, CS, FA, GSA and JADE methods, while there is no
significant difference between SMS and L-SHADE or between SMS and PSO. Since the SMS algorithm has a better ranking than the L-SHADE and PSO algorithms, the SMS algorithm is a better choice to train the DNN model. In summary, the SMS algorithm shows obvious advantages over other optimization algorithms in training the DNN for the daily PM$_{2.5}$ concentration prediction.

Table 6. Statistical analysis of the SMS algorithm for the PM$_{2.5}$ concentration time series compared to other heuristic optimization algorithms.

| Algorithm | Ranking | $z$-Value | Unadjusted $p$ | $p_{bonf}$ |
|-----------|---------|-----------|----------------|------------|
| GA        | 5.5     | 2.9462    | 0.003216       | 0.02251    |
| CS        | 6.0833  | 3.3588    | 0.000783       | 0.00548    |
| FA        | 5.5833  | 3.0052    | 0.002654       | 0.01858    |
| GSA       | 7.8333  | 4.5962    | 0.0000004      | 0.00003    |
| JADE      | 4.5     | 2.2392    | 0.025145       | 0.09601    |
| L-SHADE   | 1.6667  | 0.2357    | 0.813664       | 5.69564    |
| PSO       | 3.5     | 1.5321    | 0.125506       | 0.87855    |
| SMS       | 1.3333  | -         | -              | -          |

4.5.2. Comparison with Other Prediction Approaches

The above experimental results show that the SMS algorithm is a promising learning algorithm for optimizing the SDNN with less prediction error and more stability than the other methods. We also compare the SDNN to eight other commonly applied prediction models: the multilayer perceptron (MLP) [55], classic DNN trained by the BP algorithm (DNN-BP), S-DNN, decision tree (DT) model, SVR with a linear kernel (SVR-L), SVR with a polynomial kernel (SVR-P), SVR with a radial basis function kernel (SVR-R), and LSTM model. For a fair comparison, the high parameters of all DNN-related models are determined by Taguchi’s method in accordance with the SDNN. The initial hyperparameters of these prediction models for each dataset are presented in Table 7.

Table 7. Parameter values of the prediction models for the PM$_{2.5}$ concentration time series data.

| Models | Parameter | Value |
|--------|-----------|-------|
| MLP    | Hidden layer number | 4, 7, 13, 4, 4, 10 |
| DNN-BP | $K$ | 18, 14, 6, 18, 18 |
|        | $M$ | 4, 7, 13, 4, 10 |
|        | $\beta$ | 1.2, 0.9, 1.2, 1.2, 1.2, 0.9 |
| S-DNN  | $K$ | 18, 14, 6, 18 |
|        | $\beta$ | 1.2, 0.9, 1.2, 1.2, 1.2, 0.9 |
| DT     | Minleaf | 25 |
|        | Maxleaf and Maxdepth | Default |
|        | Cost ($c$) | 0.5, 0.5, 0.5 |
|        | Epsilon of loss function ($\epsilon$) | 0.01, 0.2, 0.01 |
|        | $\gamma$ | 1/5 |
| SVR-L, SVR-P, SVR-R | Cost ($c$) | 0.5, 0.5, 0.5 |
| LSTM   | Hidden units | 200 |
|        | Maximum epochs | 1000 |

Based on PSR, the six one-dimensional PM$_{2.5}$ concentration time series data independently transform into six high-dimensional training datasets, which are input into the SDNN for training. Figure 5 (left) shows the corresponding forecast PM$_{2.5}$ concentration obtained after the training process compared to the monitoring value, where the black and light blue lines represent the observed and predicted PM$_{2.5}$ concentrations, respectively. The observed and predicted time series are relatively close for each dataset. In addition, to examine the correlation between the observed and predicted data, scatter plots are
shown in Figure 5 (right). The figure demonstrates that the distribution of the points approximately converges very near the regression line for all PM$_{2.5}$ concentration data. Notably, the SDNN fails at a few valley and peak values, which can be confirmed from these scatter plots. Thus, the SDNN must still be improved to avoid overestimating or underestimating lower or higher PM$_{2.5}$ concentrations during air quality forecasting.

Figure 5. PM$_{2.5}$ concentration time series training and prediction results obtained by the SDNN.

To further verify the superiority of the SDNN in forecasting PM$_{2.5}$ concentrations, a quantitative evaluation is performed. The SDNN is compared to the MLP, DNN-BP, S-DNN, DT, LSTM and SVR models with three different kernels. The overall performances of the prediction models, which are measured by the average value of five estimation metrics for 30 repeated experiments, are summarized in Tables 8 and 9. The optimal values are marked in bold. To detect significant differences between the SDNN and the other prediction models, the Wilcoxon signed-rank test, which is a nonparametric statistical test, is employed in this section. The $p$-values are calculated and presented on the right of each evaluation metric in Tables 8 and 9, where “-” denotes “not applicable”. The significance
level is set to 0.05 [56], which indicates that if the p-value exceeds 0.05, there is no significant difference between the two compared models. Otherwise, there are significant advantages over the competitor.

As illustrated in Tables 8 and 9, the MSE, MAE and CE of the SDNN are clearly better than those of the other prediction approaches for all datasets. The corresponding p-values imply that on most of these evaluation metrics, the SDNN and its competitors show significant differences. The better forecasting performance of the proposed SDNN is thus evident. With respect to MAPE and RMSE, the SDNN performs better than the MLP, DNN-BP, S-DNN, SVR-P, SVR-R and LSTM methods for most of the prediction datasets. However, it performs worse than the DT and SVR-L methods. Specifically, the MAPE of the DT model shows the best results for 5 (of 6) datasets and the RMSE of SVR-L shows the best results for 3 (of 6) datasets. Surprisingly, these two relatively simple machine learning techniques perform better than other more complex approaches such as the LSTM and SDNN methods. The advantages of the SDNN here are not significant and can be the reason for the low smoothness levels in airborne pollution.

In general, from the results in Tables 8 and 9, the SDNN shows obvious advantages in terms of PM$_{2.5}$ concentration prediction. The SDNN is more stable and robust than other prediction approaches, since the SDNN complex dendritic structure can more deeply and effectively extract useful feature information and nonlinear relationships between distinct features of the input datasets than its competitors. To better demonstrate the integrated capabilities of SDNN, including the MSE, MAE, MAPE, and RMSE values, the error stacked bars are plotted in Figure 6. The SDNN achieves a lower error column than the other eight models for all PM$_{2.5}$ concentration datasets. This result confirms that the proposed SDNN exhibits effective predictive performance and strong robustness. According to the above experimental results, compared to other prediction models, the SDNN achieves very competitive forecasting performance and can be considered an efficient and effective PM$_{2.5}$ concentration forecasting approach. In addition, the CEs of all models are not ideal (less than 0.8), so there is still much room to improve the forecasting performance of machine learning approaches.

Figure 6. Comparison of the four error criteria for the six PM$_{2.5}$ concentration time series.
Table 8. Prediction performances of all the models for PM$_{2.5}$ concentration time series datasets 1–3.

| Models    | MSE (Mean ± Std) | $p$-value | MAE (Mean ± Std) | $p$-value | MAPE (Mean ± Std) | $p$-value | RMSE (Mean ± Std) | $p$-value | CE (Mean ± Std) |
|-----------|------------------|-----------|-------------------|-----------|-------------------|-----------|-------------------|-----------|-----------------|
| MLP       | $6.54 \times 10^{-3}$ ± $9.38 \times 10^{-4}$ | $9.13 \times 10^{-7}$ | $6.73 \times 10^{-2}$ ± $6.05 \times 10^{-3}$ | $1.24 \times 10^{-6}$ | $6.15 \times 10^{-4}$ ± $3.73 \times 10^{-4}$ | $8.43 \times 10^{-1}$ | $8.07 \times 10^{-2}$ ± $5.74 \times 10^{-3}$ | $9.13 \times 10^{-7}$ | $6.10 \times 10^{-1}$ ± $1.31 \times 10^{-1}$ |
| DNN-BP    | $1.75 \times 10^{-2}$ ± $3.89 \times 10^{-7}$ | $9.13 \times 10^{-7}$ | $1.07 \times 10^{-1}$ ± $1.86 \times 10^{-6}$ | $9.13 \times 10^{-7}$ | $5.56 \times 10^{-3}$ ± $6.18 \times 10^{-8}$ | $9.13 \times 10^{-7}$ | $1.32 \times 10^{-1}$ ± $1.47 \times 10^{-6}$ | $9.13 \times 10^{-7}$ | $5.42 \times 10^{-1}$ ± $2.60 \times 10^{-1}$ |
| S-DNN     | $2.16 \times 10^{-2}$ ± $3.64 \times 10^{-2}$ | $1.24 \times 10^{-6}$ | $1.09 \times 10^{-1}$ ± $6.77 \times 10^{-2}$ | $1.24 \times 10^{-6}$ | $4.84 \times 10^{-3}$ ± $2.62 \times 10^{-3}$ | $2.04 \times 10^{-6}$ | $1.32 \times 10^{-1}$ ± $6.62 \times 10^{-2}$ | $1.24 \times 10^{-6}$ | $4.98 \times 10^{-1}$ ± $2.57 \times 10^{-1}$ |
| DT        | $6.13 \times 10^{-3}$ ± $1.76 \times 10^{-18}$ | $1.01 \times 10^{-6}$ | $5.88 \times 10^{-2}$ ± $2.82 \times 10^{-17}$ | $1.01 \times 10^{-6}$ | $7.73 \times 10^{-4}$ ± $2.21 \times 10^{-19}$ | $1.01 \times 10^{-6}$ | $7.83 \times 10^{-2}$ ± $0.00 \times 10^{0}$ | $1.01 \times 10^{-6}$ | $5.43 \times 10^{-1}$ ± $2.82 \times 10^{-16}$ |
| SVR-L     | $5.45 \times 10^{-3}$ ± $8.82 \times 10^{-19}$ | $1.63 \times 10^{-5}$ | $6.12 \times 10^{-2}$ ± $4.23 \times 10^{-17}$ | $9.13 \times 10^{-7}$ | $6.31 \times 10^{-4}$ ± $1.10 \times 10^{-19}$ | $1.16 \times 10^{-1}$ | $7.38 \times 10^{-2}$ ± $2.82 \times 10^{-17}$ | $1.63 \times 10^{-5}$ | $5.91 \times 10^{-1}$ ± $2.26 \times 10^{-16}$ |
| SVR-P     | $7.40 \times 10^{-3}$ ± $1.00 \times 10^{0}$ | $9.13 \times 10^{-7}$ | $7.33 \times 10^{-2}$ ± $9.00 \times 10^{0}$ | $9.13 \times 10^{-7}$ | $6.07 \times 10^{-4}$ ± $1.10 \times 10^{-19}$ | $1.21 \times 10^{-1}$ | $8.60 \times 10^{-2}$ ± $1.41 \times 10^{-17}$ | $9.13 \times 10^{-1}$ | $4.53 \times 10^{-1}$ ± $2.26 \times 10^{-16}$ |
| SVR-R     | $5.36 \times 10^{-3}$ ± $2.65 \times 10^{-18}$ | $1.63 \times 10^{-5}$ | $6.08 \times 10^{-2}$ ± $3.53 \times 10^{-17}$ | $9.13 \times 10^{-7}$ | $6.06 \times 10^{-4}$ ± $5.51 \times 10^{-19}$ | $1.01 \times 10^{-2}$ | $7.32 \times 10^{-2}$ ± $2.82 \times 10^{-17}$ | $1.63 \times 10^{-5}$ | $7.12 \times 10^{-1}$ ± $2.82 \times 10^{-16}$ |
| LSTM      | $5.53 \times 10^{-3}$ ± $1.95 \times 10^{-5}$ | $2.47 \times 10^{-5}$ | $5.93 \times 10^{-2}$ ± $1.25 \times 10^{-3}$ | $8.87 \times 10^{-6}$ | $8.98 \times 10^{-4}$ ± $9.79 \times 10^{-5}$ | $2.01 \times 10^{-3}$ | $7.73 \times 10^{-2}$ ± $1.42 \times 10^{-3}$ | $2.11 \times 10^{-5}$ | $7.45 \times 10^{-1}$ ± $2.37 \times 10^{-2}$ |
| SDNN      | $5.05 \times 10^{-3}$ ± $6.37 \times 10^{-5}$ | - | $5.70 \times 10^{-2}$ ± $6.14 \times 10^{-4}$ | - | $5.76 \times 10^{-4}$ ± $4.45 \times 10^{-5}$ | - | $7.13 \times 10^{-2}$ ± $1.62 \times 10^{-3}$ | - | $7.92 \times 10^{-1}$ ± $1.35 \times 10^{-2}$ |

PM$_{2.5}$ concentration data 2

| Models    | MSE (Mean ± Std) | $p$-value | MAE (Mean ± Std) | $p$-value | MAPE (Mean ± Std) | $p$-value | RMSE (Mean ± Std) | $p$-value | CE (Mean ± Std) |
|-----------|------------------|-----------|-------------------|-----------|-------------------|-----------|-------------------|-----------|-----------------|
| MLP       | $8.51 \times 10^{-3}$ ± $6.40 \times 10^{-4}$ | $1.78 \times 10^{-5}$ | $7.16 \times 10^{-2}$ ± $3.51 \times 10^{-3}$ | $1.01 \times 10^{-6}$ | $1.49 \times 10^{-3}$ ± $1.48 \times 10^{-4}$ | $2.79 \times 10^{-2}$ | $9.22 \times 10^{-2}$ ± $3.46 \times 10^{-3}$ | $1.63 \times 10^{-5}$ | $3.37 \times 10^{-1}$ ± $1.07 \times 10^{-1}$ |
| DNN-BP    | $4.83 \times 10^{-2}$ ± $1.40 \times 10^{-1}$ | $9.13 \times 10^{-7}$ | $1.42 \times 10^{-1}$ ± $1.40 \times 10^{-1}$ | $9.13 \times 10^{-7}$ | $6.20 \times 10^{-3}$ ± $3.20 \times 10^{-4}$ | $9.13 \times 10^{-7}$ | $1.75 \times 10^{-1}$ ± $1.35 \times 10^{-1}$ | $9.13 \times 10^{-7}$ | $3.41 \times 10^{-1}$ ± $2.04 \times 10^{-1}$ |
| S-DNN     | $2.26 \times 10^{-2}$ ± $2.28 \times 10^{-4}$ | $9.13 \times 10^{-7}$ | $1.16 \times 10^{-1}$ ± $2.57 \times 10^{-4}$ | $9.13 \times 10^{-7}$ | $5.59 \times 10^{-3}$ ± $9.26 \times 10^{-5}$ | $9.13 \times 10^{-7}$ | $1.50 \times 10^{-1}$ ± $7.63 \times 10^{-4}$ | $9.13 \times 10^{-7}$ | $4.98 \times 10^{-1}$ ± $1.66 \times 10^{-1}$ |
| DT        | $1.12 \times 10^{-2}$ ± $3.53 \times 10^{-18}$ | $9.13 \times 10^{-7}$ | $7.50 \times 10^{-2}$ ± $5.65 \times 10^{-17}$ | $9.13 \times 10^{-7}$ | $8.18 \times 10^{-4}$ ± $4.41 \times 10^{-19}$ | $1.00 \times 10^{0}$ | $1.06 \times 10^{-1}$ ± $2.82 \times 10^{-17}$ | $9.13 \times 10^{-7}$ | $5.09 \times 10^{-1}$ ± $2.82 \times 10^{-16}$ |
Table 8. Cont.

| Models | MSE (Mean ± Std) | p-value | MAE (Mean ± Std) | p-value | MAPE (Mean ± Std) | p-value | RMSE (Mean ± Std) | p-value | CE (Mean ± Std) |
|--------|------------------|---------|------------------|---------|-------------------|---------|--------------------|---------|-----------------|
| MLP    | 1.06 ± 1.28      | 1.13 × 10^-7 | 7.54 ± 4.87      | 1.13 × 10^-7 | 1.60 ± 2.01      | 1.03 × 10^-5 | 8.76 ± 5.24       | 1.42 × 10^-5 | 9.13 ± 4.30     |
| DNN-BP | 2.34 ± 3.51      | 1.13 × 10^-7 | 1.10 ± 2.16      | 1.01 × 10^-6 | 4.71 ± 1.68      | 1.85 × 10^-6 | 1.50 ± 1.50       | 2.84 × 10^-2 | 9.13 ± 1.15     |
| S-DNN  | 1.25 ± 1.55      | 1.13 × 10^-7 | 8.39 ± 1.89      | 1.93 ± 1.23  | 1.73 ± 1.57      | 1.10 × 10^-1 | 2.07 ± 2.07       | 1.41 × 10^-1 | 9.13 ± 4.99     |
| DT     | 1.15 ± 1.00      | 1.13 × 10^-7 | 7.41 ± 1.41      | 1.93 ± 1.41  | 2.00 ± 1.00      | 1.00 × 10^0  | 8.66 ± 1.96       | 1.39 × 10^-2 | 9.13 ± 4.17     |
| SVR-L  | 9.95 ± 1.32      | 1.13 × 10^-7 | 7.55 ± 4.23      | 1.13 × 10^-7 | 1.45 ± 8.82      | 3.66 × 10^-6 | 9.98 ± 2.66       | 1.41 × 10^-1 | 9.13 ± 5.56     |
| SVR-P  | 1.30 ± 1.76      | 1.13 × 10^-7 | 8.62 ± 4.23      | 1.13 × 10^-7 | 1.93 ± 8.82      | 3.66 × 10^-6 | 9.98 ± 2.66       | 1.41 × 10^-1 | 9.13 ± 5.56     |
| SVR-R  | 9.77 ± 1.33      | 1.13 × 10^-7 | 7.50 ± 4.23      | 1.13 × 10^-7 | 1.45 ± 8.82      | 3.66 × 10^-6 | 9.98 ± 2.66       | 1.41 × 10^-1 | 9.13 ± 5.56     |
| LSTM   | 8.55 ± 1.17      | 1.13 × 10^-7 | 6.46 ± 5.03      | 1.17 × 10^-3 | 1.46 ± 1.50      | 3.92 × 10^-2 | 9.42 ± 2.94       | 1.98 × 10^-3 | 9.13 ± 5.56     |
| SDNN   | 7.71 ± 1.77      | 1.13 × 10^-7 | 6.46 ± 1.16      | 1.35 ± 6.99  | 1.84 ± 1.25      | 1.98 × 10^-3 | 9.42 ± 2.94       | 1.98 × 10^-3 | 9.13 ± 5.56     |

PM$_{2.5}$ concentration data

- 3
Table 9. Prediction performances of all the models for PM$_{2.5}$ concentration time series datasets 4–6.

| PM$_{2.5}$ concentration data 4                  | Models  | MSE (Mean ± Std) | p-value | MAE (Mean ± Std) | p-value | MAPE (Mean ± Std) | p-value | RMSE (Mean ± Std) | p-value | CE (Mean ± Std) |
|------------------------------------------------|---------|------------------|---------|------------------|---------|-------------------|---------|-------------------|---------|-----------------|
| MLP                                            | 1.97 × 10$^{-2}$ ± 2.00 × 10$^{-3}$ | 1.37 × 10$^{-6}$ | 1.06 × 10$^{-1}$ ± 4.59 × 10$^{-3}$ | 9.13 × 10$^{-7}$ | 1.67 × 10$^{-3}$ ± 1.71 × 10$^{-4}$ | 1.24 × 10$^{-6}$ | 1.40 × 10$^{-1}$ ± 7.11 × 10$^{-3}$ | 3.30 × 10$^{-5}$ | 4.78 × 10$^{-1}$ ± 1.23 × 10$^{-1}$ |
| DNN-BP                                         | 6.13 × 10$^{-2}$ ± 5.40 × 10$^{-6}$ | 9.13 × 10$^{-7}$ | 1.96 × 10$^{-1}$ ± 1.52 × 10$^{-5}$ | 9.13 × 10$^{-7}$ | 5.57 × 10$^{-3}$ ± 2.47 × 10$^{-7}$ | 9.13 × 10$^{-7}$ | 2.48 × 10$^{-1}$ ± 1.09 × 10$^{-5}$ | 9.13 × 10$^{-7}$ | 3.68 × 10$^{-2}$ ± 1.94 × 10$^{-1}$ |
| S-DNN                                          | 4.58 × 10$^{-2}$ ± 1.18 × 10$^{-1}$ | 8.59 × 10$^{-6}$ | 1.38 × 10$^{-1}$ ± 1.30 × 10$^{-1}$ | 5.91 × 10$^{-6}$ | 2.52 × 10$^{-3}$ ± 2.39 × 10$^{-3}$ | 3.53 × 10$^{-3}$ | 1.73 × 10$^{-1}$ ± 1.28 × 10$^{-1}$ | 9.13 × 10$^{-7}$ | 5.68 × 10$^{-2}$ ± 1.94 × 10$^{-2}$ |
| DT                                             | 3.44 × 10$^{-2}$ ± 2.12 × 10$^{-17}$ | 9.13 × 10$^{-7}$ | 1.27 × 10$^{-1}$ ± 9.00 × 10$^{-9}$ | 9.13 × 10$^{-7}$ | 8.43 × 10$^{-4}$ ± 0.00 × 10$^{-9}$ | 1.00 × 10$^{-4}$ | 1.86 × 10$^{-1}$ ± 0.00 × 10$^{-9}$ | 9.13 × 10$^{-7}$ | 4.31 × 10$^{-1}$ ± 1.69 × 10$^{-16}$ |
| SVR-L                                          | 1.67 × 10$^{-2}$ ± 1.06 × 10$^{-17}$ | 3.79 × 10$^{-1}$ | 9.70 × 10$^{-2}$ ± 4.23 × 10$^{-17}$ | 1.62 × 10$^{-2}$ | 1.57 × 10$^{-3}$ ± 1.10 × 10$^{-18}$ | 9.13 × 10$^{-7}$ | 1.28 × 10$^{-1}$ ± 0.00 × 10$^{-9}$ | 1.00 × 10$^{0}$ | 5.48 × 10$^{-1}$ ± 3.39 × 10$^{-16}$ |
| SVR-P                                          | 2.24 × 10$^{-2}$ ± 1.06 × 10$^{-17}$ | 9.13 × 10$^{-7}$ | 1.09 × 10$^{-1}$ ± 7.06 × 10$^{-17}$ | 9.13 × 10$^{-7}$ | 2.40 × 10$^{-3}$ ± 4.41 × 10$^{-19}$ | 9.13 × 10$^{-7}$ | 1.50 × 10$^{-1}$ ± 5.65 × 10$^{-17}$ | 1.51 × 10$^{-6}$ | 4.34 × 10$^{-1}$ ± 0.00 × 10$^{0}$ |
| SVR-R                                          | 1.68 × 10$^{-2}$ ± 3.53 × 10$^{-18}$ | 3.56 × 10$^{-1}$ | 9.77 × 10$^{-2}$ ± 4.23 × 10$^{-17}$ | 3.71 × 10$^{-4}$ | 1.60 × 10$^{-3}$ ± 0.00 × 10$^{0}$ | 9.13 × 10$^{-7}$ | 1.29 × 10$^{-1}$ ± 5.65 × 10$^{-17}$ | 1.00 × 10$^{0}$ | 5.55 × 10$^{-1}$ ± 1.13 × 10$^{-16}$ |
| LSTM                                           | 4.13 × 10$^{-2}$ ± 4.98 × 10$^{-2}$ | 9.13 × 10$^{-7}$ | 1.36 × 10$^{-1}$ ± 5.51 × 10$^{-2}$ | 9.13 × 10$^{-7}$ | 1.53 × 10$^{-3}$ ± 5.19 × 10$^{-4}$ | 9.13 × 10$^{-7}$ | 1.89 × 10$^{-1}$ ± 7.43 × 10$^{-2}$ | 9.13 × 10$^{-7}$ | 5.55 × 10$^{-1}$ ± 1.13 × 10$^{-17}$ |
| SDNN                                           | 1.67 × 10$^{-2}$ ± 7.60 × 10$^{-4}$ | - | 9.64 × 10$^{-2}$ ± 1.67 × 10$^{-3}$ | - | 9.37 × 10$^{-4}$ ± 4.37 × 10$^{-5}$ | - | 1.31 × 10$^{-1}$ ± 7.50 × 10$^{-3}$ | - | 5.98 × 10$^{-1}$ ± 2.03 × 10$^{-2}$ |

| PM$_{2.5}$ concentration data 5                  | Models  | MSE (Mean ± Std) | p-value | MAE (Mean ± Std) | p-value | MAPE (Mean ± Std) | p-value | RMSE (Mean ± Std) | p-value | CE (Mean ± Std) |
|------------------------------------------------|---------|------------------|---------|------------------|---------|-------------------|---------|-------------------|---------|-----------------|
| MLP                                            | 1.36 × 10$^{-2}$ ± 1.50 × 10$^{-3}$ | 1.37 × 10$^{-6}$ | 9.01 × 10$^{-2}$ ± 5.16 × 10$^{-3}$ | 9.13 × 10$^{-7}$ | 1.18 × 10$^{-3}$ ± 2.16 × 10$^{-4}$ | 1.83 × 10$^{-1}$ | 1.16 × 10$^{-1}$ ± 5.54 × 10$^{-3}$ | 1.37 × 10$^{-6}$ | 3.39 × 10$^{-1}$ ± 1.19 × 10$^{-1}$ |
| DNN-BP                                         | 1.07 × 10$^{-1}$ ± 2.07 × 10$^{-1}$ | 9.13 × 10$^{-7}$ | 2.25 × 10$^{-1}$ ± 2.09 × 10$^{-1}$ | 9.13 × 10$^{-7}$ | 6.59 × 10$^{-3}$ ± 3.42 × 10$^{-3}$ | 9.13 × 10$^{-7}$ | 2.62 × 10$^{-1}$ ± 1.99 × 10$^{-1}$ | 9.13 × 10$^{-7}$ | 4.07 × 10$^{-1}$ ± 2.03 × 10$^{-1}$ |
| S-DNN                                          | 4.05 × 10$^{-2}$ ± 1.73 × 10$^{-5}$ | 9.13 × 10$^{-7}$ | 1.61 × 10$^{-1}$ ± 6.71 × 10$^{-5}$ | 9.13 × 10$^{-7}$ | 5.59 × 10$^{-3}$ ± 1.20 × 10$^{-6}$ | 9.13 × 10$^{-7}$ | 2.01 × 10$^{-1}$ ± 4.31 × 10$^{-5}$ | 9.13 × 10$^{-7}$ | 4.22 × 10$^{-2}$ ± 1.47 × 10$^{-1}$ |
| DT                                             | 1.39 × 10$^{-2}$ ± 8.82 × 10$^{-18}$ | 9.13 × 10$^{-7}$ | 9.20 × 10$^{-2}$ ± 5.65 × 10$^{-17}$ | 9.13 × 10$^{-7}$ | 8.40 × 10$^{-4}$ ± 2.21 × 10$^{-19}$ | 1.00 × 10$^{0}$ | 1.18 × 10$^{-1}$ ± 7.06 × 10$^{-17}$ | 9.13 × 10$^{-7}$ | 4.69 × 10$^{-1}$ ± 1.69 × 10$^{-16}$ |
### Table 9. Cont.

| Models   | MSE (Mean ± Std) | p-value | MAE (Mean ± Std) | p-value | MAPE (Mean ± Std) | p-value | RMSE (Mean ± Std) | p-value | CE (Mean ± Std) |
|----------|------------------|---------|------------------|---------|-------------------|---------|-------------------|---------|-----------------|
| SVR-L    | 1.18 × 10⁻² ± 1.76 × 10⁻¹² | 1.00 × 10⁰ | 7.89 × 10⁻² ± 7.06 × 10⁻¹⁷ | 1.63 × 10⁻⁵ | 1.18 × 10⁻³ ± 0.00 × 10⁰ | 2.48 × 10⁻⁶ | 1.07 × 10⁻¹ ± 5.65 × 10⁻¹⁷ | 1.00 × 10⁰ | 4.87 × 10⁻¹ ± 2.26 × 10⁻¹⁶ |
| SVR-P    | 1.40 × 10⁻² ± 3.53 × 10⁻¹⁸ | 9.13 × 10⁻⁷ | 8.99 × 10⁻² ± 0.00 × 10⁰ | 9.13 × 10⁻⁷ | 1.85 × 10⁻³ ± 4.41 × 10⁻¹⁹ | 9.13 × 10⁻⁷ | 1.18 × 10⁻¹ ± 2.82 × 10⁻¹⁷ | 9.13 × 10⁻⁷ | 5.38 × 10⁻¹ ± 5.65 × 10⁻¹⁷ |
| SVR-R    | 1.17 × 10⁻² ± 0.00 × 10⁰ | 1.01 × 10⁻³ | 7.89 × 10⁻² ± 1.41 × 10⁻¹⁷ | 1.63 × 10⁻⁵ | 1.10 × 10⁻³ ± 4.41 × 10⁻¹⁹ | 1.00 × 10⁰ | 1.07 × 10⁻¹ ± 1.41 × 10⁻¹⁷ | 1.00 × 10⁰ | 4.89 × 10⁻¹ ± 2.26 × 10⁻¹⁶ |
| LSTM     | 1.30 × 10⁻² ± 2.17 × 10⁻³ | 8.01 × 10⁻³ | 8.33 × 10⁻² ± 6.56 × 10⁻⁳ | 6.63 × 10⁻⁶ | 1.20 × 10⁻³ ± 3.63 × 10⁻⁴ | 1.79 × 10⁻² | 1.14 × 10⁻¹ ± 9.25 × 10⁻³ | 9.13 × 10⁻⁷ | 4.89 × 10⁻¹ ± 2.26 × 10⁻¹⁷ |
| SDNN     | 1.16 × 10⁻² ± 1.73 × 10⁻⁴ | -       | 7.73 × 10⁻² ± 1.08 × 10⁻³ | -       | 1.14 × 10⁻³ ± 2.70 × 10⁻⁵ | -       | 1.08 × 10⁻¹ ± 7.61 × 10⁻⁴ | -       | 5.97 × 10⁻¹ ± 9.69 × 10⁻³ |

**PM2.5 concentration data 6**
5. Extension

As presented above, the proposed SDNN can successfully predict PM$_{2.5}$ concentration. In this section, the performance of the proposed algorithm is evaluated on an open available PM$_{2.5}$ dataset from UCI machine learning repository (https://archive.ics.uci.edu/ml/datasets.php, accessed on 1 January 2021), which is the hourly PM$_{2.5}$ concentration of US Embassy in Beijing and the meteorological data from Beijing Capital International Airport. In addition, we compare the predictive performance of SDNN with four other prediction approaches in the literature.

Table 10 summarizes the comparison between the SDNN and other prediction techniques on the hourly PM$_{2.5}$ concentration prediction in terms of RMSE and MAE. In order to further intuitively compare these two evaluation metrics, both the RMSE and MAE are performed the operation of inverse normalization. The best results of the prediction model are highlighted in bold, and all values are the average of the experimental results. It can be observed that our proposed SDNN obtains the best result on the UCI hourly PM$_{2.5}$ concentration time series datasets, the performance of the SDNN the ranks first among five prediction techniques. Accordingly, it can be concluded that the overall performance of the SDNN is evidently better than those of other prediction models.

Table 10. Prediction performances of all the models for UCI hourly PM$_{2.5}$ concentration time series datasets.

| Model       | RMSE      | MAE       |
|-------------|-----------|-----------|
| CNN-LSTM [57] | 18.0852   | 15.3243   |
| CBGRU [58]   | 14.5319   | 10.4798   |
| SVM-GA [59]  | 26.4066   | -         |
| MKL [60]     | -         | 12.90     |
| SDNN         | 12.9188   | 10.4513   |

6. Conclusions

Predicting the air quality is beneficial for the protection, early monitoring and governance of the environment. However, due to the characteristics of PM$_{2.5}$ motion, it is difficult to predict PM$_{2.5}$ concentrations with high accuracy and stability. In this paper, a novel SDNN is proposed to improve the accuracy of PM$_{2.5}$ concentration time series forecasting. The proposed SDNN is trained by the SMS global optimization algorithm due to its powerful search abilities. To evaluate the effectiveness of the SDNN, six prediction datasets are adopted in our experiments. The MI and FNN approaches are employed to obtain the time delay and embedding dimensions, respectively. Then, the phase space is reconstructed based on these two factors, and the MLE is used to analyse the predictable limit and chaotic characteristics of PM$_{2.5}$ concentration datasets. Finally, the prediction results of the SDNN are tested for the regenerated datasets and compared to those of DNNs trained by six optimization algorithms and eight commonly used prediction models. The experimental results and statistical analysis demonstrate that the SDNN dominates in terms of the four evaluation metrics. Thus, the proposed model can effectively enhance the stability and accuracy of PM$_{2.5}$ concentration predictions. While the SDNN achieves more competitive forecasting results, there is still much room to improve the forecasting performance of machine learning approaches in terms of CE results. While this study employs only historical PM$_{2.5}$ concentrations as an influencing factor, more auxiliary information, such as weather conditions, economic factors and geographical positions, will be considered in our future study. In addition, the SDNN must be applied to solve other real-world time series prediction problems such as those of traffic flow forecasting and financial time series prediction.

Author Contributions: Conceptualization, Z.S. and J.J.; methodology, Z.S.; software, C.T. and Y.T.; validation, Y.T. and Z.T.; formal analysis, Z.S.; resources, Z.T.; writing—original draft preparation, Z.S.; writing—review and editing, J.J. and Y.T.; visualization, Z.S. and C.T. All authors have read and agreed to the published version of the manuscript.
Funding: This study was partially supported by the Nature Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 19KJ520015), the Talent Development Project of Taizhou University (No. TZXY2018QDJJ006), the Guangdong Basic and Applied Basic Research Fund Project (No. 2019A1515111139), and the National Science Foundation for Young Scientists of China (Grant No. 61802274).

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Gan, K.; Sun, S.; Wang, S.; Wei, Y. A secondary-decomposition-ensemble learning paradigm for forecasting PM$_{2.5}$ concentration. Atmos. Pollut. Res. 2018, 9, 989–999. [CrossRef]
2. Xu, Y.; Yang, W.; Wang, J. Air quality early-warning system for cities in China. Atmos. Environ. 2017, 148, 239–257. [CrossRef]
3. Agarwal, S.; Sharma, S.; Suresh, R.; Rahman, M.; H.; Vrancks, S.; Maiheu, B.; Batra, S. Air quality forecasting using artificial neural networks with real time dynamic error correction in highly polluted regions. Sci. Total. Environ. 2020, 735, 139454. [CrossRef]
4. Cekim, H.O. Forecasting PM10 concentrations using time series models: A case of the most polluted cities in Turkey. Environ. Sci. Pollut. Res. Int. 2020, 27, 25612–25624. [CrossRef]
5. Wang, D.; Wei, S.; Luo, H.; Yue, C.; Grunder, O. A novel hybrid model for air quality index forecasting based on two-phase decomposition technique and modified extreme learning machine. Sci. Total. Environ. 2017, 580, 719–733. [CrossRef]
6. Lv, B.; Cobourn, W.G.; Bai, Y. Development of nonlinear empirical models to forecast daily PM$_{2.5}$ and ozone levels in three large Chinese cities. Atmos. Environ. 2016, 147, 209–223. [CrossRef]
7. Sahu, R.; Nagal, A.; Dixit, K.K.; Unnibhavi, H.; Mantravadi, S.; Nair, S.; Tripathi, S.N. Robust statistical calibration and characterization of portable low-cost air quality monitoring sensors to quantify real-time O$_3$ and NO$_2$ concentrations in diverse environments. Atmos. Meas. Tech. 2021, 14, 37–52. [CrossRef]
8. Stafoggia, M.; Bellander, T.; Bucci, S.; Davoli, M.; De Hoogh, K.; De’Donato, F.; Scortichini, M. Estimation of daily PM$_{10}$ and PM$_{2.5}$ concentrations in Italy, 2013–2015. using a spatiotemporal land-use random-forest model. Environ. Int. 2019, 124, 170–179. [CrossRef][PubMed]
9. Zhou, Y.; Chang, F.; Chang, L.C.; Kao, I.F.; Wang, Y.S.; Kang, C.C. Multi-output support vector machine for regional multi-step-ahead PM$_{2.5}$ forecasting. Sci. Total. Environ. 2019, 651, 230–240. [CrossRef]
10. Todo, Y.; Tamura, H.; Yamashita, K.; Tang, Z. Unsupervised learnable neuron model with nonlinear interaction on dendrites. Neural Netw. 2014, 60, 96–103. [CrossRef]
11. Taylor, W.R.; He, S.; Levick, W.R.; Vaney, D.I. Dendritic computation of direction selectivity by retinal ganglion cells. Science 2000, 289, 2347–2350. [CrossRef]
12. Tang, C.; Ji, J.; Tang, Y.; Gao, S.; Tang, Z.; Todo, Y. A novel machine learning technique for computer-aided diagnosis. Eng. Appl. Artif. Intell. 2020, 92, 103627. [CrossRef]
13. Song, Z.; Tang, Y.; Ji, J.; Todo, Y. Evaluating a dendritic neuron model for wind speed forecasting. Knowl. Based Syst. 2020, 201, 106052. [CrossRef]
14. Song, S.; Chen, X.; Tang, C.; Song, S.; Tang, Z.; Todo, Y. Training an Approximate Logic Dendritic Neuron Model Using Social Learning Particle Swarm Optimization Algorithm. IEEE Access 2019, 7, 141947–141959. [CrossRef]
15. Cuevas, E.; Echavarria, A.; Ramirez-Ortega, M.A. An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. Appl. Intell. 2014, 40, 256–272. [CrossRef]
16. Takens, F. Detecting strange attractors in turbulence. In Dynamical Systems and Turbulence; Springer: Warwick, UK, 1980; pp. 366–381.
17. Fraser, A.M.; Swinney, H.L. Independent coordinates for strange attractors from mutual information. Phys. Rev. A 1986, 33, 1134. [CrossRef]
18. Kennel, M.B.; Brown, R.; Abarbanel, H.D. Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys. Rev. A 1992, 45, 3403. [CrossRef][PubMed]
19. Kantz, H. A robust method to estimate the maximal Lyapunov exponent of a time series. Phys. Lett. A 1994, 185, 77–87. [CrossRef]
20. Elbayoumi, M.; Ramli, N.A.; Yusof, N.F.F.M. Development and comparison of regression models and feedforward backpropagation neural network models to predict seasonal indoor PM$_{2.5}$–10 and PM$_{2.5}$ concentrations in naturally ventilated schools. Atmos. Pollut. Res. 2015, 6, 1013–1023. [CrossRef]
21. Lin, Y.C.; Lee, S.J.; Ouyang, C.S.; Wu, C.H. Air quality prediction by neuro-fuzzy modeling approach. Applied soft computing. Sensors Actuators B Chem. 2020, 86, 105898.
22. Bai, Y.; Zeng, B.; Li, C.; Zhang, J. An ensemble long short-term memory neural network for hourly PM$_{2.5}$ concentration forecasting. Chemosphere 2019, 222, 286–294. [CrossRef][PubMed]
23. Xu, X.; Yonedo, M. Multitask Air-Quality Prediction Based on LSTM-Autoencoder Model. IEEE Trans. Cybern. 2019. [CrossRef]
24. Ma, J.; Ding, Y.; Cheng, J.C.; Jiang, F.; Tan, Y.; Gan, V.J.; Wan, Z. Identification of high impact factors of air quality on a national scale using big data and machine learning techniques. J. Clean. Prod. 2020, 244, 118955. [CrossRef]
25. Loy-Benitez, J.; Vilela, P.; Li, Q.; Yoo, C. Sequential prediction of quantitative health risk assessment for the fine particulate matter in an underground facility using deep recurrent neural networks. Ecotoxicol. Environ. Saf. 2019, 169, 316–324. [CrossRef]
26. Li, T.; Shen, H.; Yuan, Q.; Zhang, X.; Zhang, L. Estimating ground-level PM$_{2.5}$ by fusing satellite and station observations: A geo-intelligent deep learning approach. *Geophys. Res. Lett.* **2017**, *44*, 11–985. [CrossRef]

27. Huang, C.J.; Kuo, P.H. A deep cnn-lstm model for particulate matter (PM$_{2.5}$) forecasting in smart cities. *Sensors* **2018**, *18*, 2220. [CrossRef]

28. Pak, U.; Ma, J.; Ryu, U.; Ryom, K.; Juhuyok, U.; Pak, K.; Pak, C. Deep learning-based PM$_{2.5}$ prediction considering the spatiotemporal correlations: A case study of Beijing, China. *Sci. Total. Environ.* **2020**, *699*, 133561. [CrossRef] [PubMed]

29. Liu, H.; Chen, C. Prediction of outdoor PM$_{2.5}$ concentrations based on a three-stage hybrid neural network model. *Atmos. Pollut. Res.* **2020**, *11*, 469–481. [CrossRef]

30. Voukantis, D.; Karatzas, K.; Kukkonen, J.; Räsänen, T.; Karppinen, A.; Kolehmainen, M. Intercomparison of air quality data using principal component analysis, and forecasting of PM$_{10}$ and PM$_{2.5}$ concentrations using artificial neural networks, in Thessaloniki and Helsinki. *Sci. Total. Environ.* **2011**, *409*, 1266–1276. [CrossRef]

31. Abderrahim, H.; Chellali, M.R.; Hamou, A. Forecasting PM$_{10}$ in Algiers: Efficacy of multilayer perceptron networks. *Environ. Sci. Pollut. Res.* **2016**, *23*, 1634–1641. [CrossRef]

32. Fu, M.; Wang, W.; Le, Z.; Khorraram, M.S. Prediction of particulate matter concentrations by developed feed-forward neural network with rolling mechanism and gray model. *Neural Comput. Appl.* **2015**, *26*, 1789–1797. [CrossRef]

33. Gao, S.; Zhao, H.; Bai, Z.; Han, B.; Xu, J.; Zhao, R.; Yu, H. Combined use of principal component analysis and artificial neural network approach to improve estimates of PM$_{2.5}$ personal exposure: A case study on older adults. *Sci. Total. Environ.* **2020**, *726*, 138533. [CrossRef] [PubMed]

34. Biancofiore, F.; Busilacchio, M.; Verdecchia, M.; Tomasetti, B.; Aruffo, E.; Bianco, S.; Di Carlo, P. Recursive neural network model for analysis and forecast of PM$_{10}$ and PM$_{2.5}$. *Atmos. Pollut. Res.* **2017**, *8*, 652–659. [CrossRef]

35. Yegeaneh, B.; Hewson, M.G.; Clifford, S.; Tavassoli, A.; Knibbs, L.D.; Morawska, L. Estimating the spatiotemporal variation of NO$_2$ concentration using an adaptive neuro-fuzzy inference system. *Environ. Model. Softw.* **2018**, *100*, 222–235. [CrossRef]

36. Feng, X.; Li, Q.; Zhu, Y.; Hou, J.; Jin, L.; Wang, J. Artificial neural networks forecasting of PM$_{2.5}$ pollution using air mass trajectory based geographic model and wavelet transformation. *Atmos. Environ.* **2015**, *107*, 118–128. [CrossRef]

37. Ordieres, J.B.; Vergara, E.P.; Capuz, R.S.; Salazar, R.E. Artificial neural network prediction model for fine particulate matter (PM$_{2.5}$) on the US-Mexico border in El Paso (Texas) and Ciudad Juárez (Chihuahua). *Environ. Model. Softw.* **2005**, *20*, 547–559. [CrossRef]

38. Sun, W.; Sun, J. Daily PM$_{2.5}$ concentration prediction based on principal component analysis and LSSVM optimized by cuckoo search algorithm. *J. Environ. Manag.* **2017**, *188*, 144–152. [CrossRef]

39. Liu, H.; Duan, Z.; Chen, C. A hybrid multi-resolution multi-objective ensemble model and its application for forecasting of daily PM$_{2.5}$ concentrations. *Inf. Sci.* **2020**, *516*, 266–292. [CrossRef]

40. Qi, Y.; Li, Q.; Karimian, H.; Liu, D. A hybrid model for spatiotemporal forecasting of PM$_{2.5}$ based on graph convolutional neural network and long short-term memory. *Sci. Total. Environ.* **2019**, *664*, 1–10. [CrossRef]

41. Kow, P.Y.; Wang, Y.S.; Zhou, Y.; Kao, I.F.; Issermann, M.; Chang, L.C.; Chang, F.J. Seamless integration of convolutional and back-propagation neural networks for regional multi-step-ahead PM$_{2.5}$ forecasting. *J. Clean. Prod.* **2020**, *261*, 121285. [CrossRef]

42. Di, Q.; Koutrakis, P.; Schwartz, J. A hybrid prediction model for PM$_{2.5}$ mass and components using a chemical transport model and land use regression. *Atmos. Environ.* **2016**, *131*, 390–399. [CrossRef]

43. Gabbiani, F.; Krapp, H.G.; Koch, C.; Laurent, G. Multiplicative computation in a visual neuron sensitive to looming. *Nature* **2002**, *418*, 320–324. [CrossRef]

44. Small, M. *Applied Nonlinear Time Series Analysis: Applications in Physics, Physiology and Finance*; World Scientific: Singapore, 2005; p. 52.

45. Wolf, A.; Swift, J.B.; Swinney, H.L.; Vastano, J.A. Determining Lyapunov exponents from a time series. *Phys. D Nonlinear Phenom.* **1985**, *16*, 285–317. [CrossRef]

46. Abarbanel, H. *Analysis of Observed Chaotic Data*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.

47. Atland, H.W. Computer-Based Robust Engineering: Essentials for DFSS. *Technometrics* **2006**, *48*, 547–559. [CrossRef]

48. Srinivas, M.; Patnaik, L.M. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans. Syst. Man, Cybern.* **1994**, *24*, 656–667. [CrossRef]

49. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC), Pietermaritzburg, South Africa, 9 December 2009; pp. 210–214.

50. Yang, X.S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78–84. [CrossRef]

51. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]

52. Zhang, J.; Sanderson, A.C. JADE: Adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **2009**, *13*, 945–958. [CrossRef]

53. Tanabe, R.; Fukunaga, A.S. Improving the search performance of SHADE using linear population size reduction. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1658–1665.

54. Bonyadi, M.R.; Michalewicz, Z. Particle swarm optimization for single objective continuous space problems: A review. *Evol. Comput.* **2017**, *25*, 1–54. [CrossRef] [PubMed]

55. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. *Learning Internal Representations by Error Propagation*; California Univ San Diego La Jolla Inst for Cognitive Science: San Diego, CA, USA, 1985.
56. García, S.; Molina, D.; Lozano, M.; Herrera, F. 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* **2009**, *15*, 617. [CrossRef]

57. Li, T.; Hua, M.; Wu, X. A hybrid CNN-LSTM model for forecasting particulate matter (PM$_{2.5}$). *IEEE Access* **2020**, *8*, 26933–26940. [CrossRef]

58. Tao, Q.; Liu, F.; Li, Y.; Sidorov, D. Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU. *IEEE Access* **2019**, *7*, 76690–76698. [CrossRef]

59. Chuentawat, R.; Kan-ngan, Y. The comparison of PM$_{2.5}$ forecasting methods in the form of multivariate and univariate time series based on support vector machine and genetic algorithm. In Proceedings of the IEEE 15th International Conference on Electrical Engineering/Electronics, Chiang Rai, Thailand, 18–21 July 2018, 572–575.

60. Xu, X. Forecasting air pollution PM$_{2.5}$ in Beijing using weather data and multiple kernel learning. *J. Forecast.* **2020**, *39*, 117–125. [CrossRef]