Research Article

Air Pollution Concentration Forecast Method Based on the Deep Ensemble Neural Network

Canyang Guo, Genggeng Liu, and Chi-Hua Chen

College of Mathematics and Computer Sciences, Fuzhou University, No. 2 Xue Yuan Road, University Town, Fuzhou, Fujian 350116, China

Correspondence should be addressed to Genggeng Liu; liugenggeng@fzu.edu.cn and Chi-Hua Chen; chihua0826@gmail.com

Received 28 August 2020; Revised 18 September 2020; Accepted 21 September 2020; Published 5 October 2020

Copyright © 2020 Canyang Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The global environment has become more polluted due to the rapid development of industrial technology. However, the existing machine learning prediction methods of air quality fail to analyze the reasons for the change of air pollution concentration because most of the prediction methods take more focus on the model selection. Since the framework of recent deep learning is very flexible, the model may be deep and complex in order to fit the dataset. Therefore, overfitting problems may exist in a single deep neural network model when the number of weights in the deep neural network model is large. Besides, the learning rate of stochastic gradient descent (SGD) treats all parameters equally, resulting in local optimal solution. In this paper, the Pearson correlation coefficient is used to analyze the inherent correlation of PM2.5 and other auxiliary data such as meteorological data, season data, and time stamp data which are applied to cluster for enhancing the performance. Extracted features are helpful to build a deep ensemble network (EN) model which combines the recurrent neural network (RNN), long short-term memory (LSTM) network, and gated recurrent unit (GRU) network to predict the PM2.5 concentration of the next hour. The weights of the submodel change with the accuracy of them in the validation set, so the ensemble has generalization ability. The adaptive moment estimation (Adam) an algorithm for stochastic optimization is used to optimize the weights instead of SGD. In order to compare the overall performance of different algorithms, the mean absolute error (MAE) and mean absolute percentage error (MAPE) are used as accuracy metrics in the experiments of this study. The experiment results show that the proposed method achieves an accuracy rate (i.e., MAE = 6.19 and MAPE = 16.20%) and outperforms the comparative models.

1. Introduction

In recent years, the rapid development of the industry is accompanied by air pollution which causes the death of 7 million people every year and attracts great attention worldwide [1, 2]. Among these air pollutants, PM2.5 can traverse the nasal passages during inhalation and reach the throat and even the lungs [3] and brings about a great threat to the human body. In 2018, Heft-Neal et al. [4] suggested that PM2.5 concentration above minimum exposure levels was responsible for 22% of infant deaths in the 30 studied countries and led to 449,000 additional deaths of infants in 2015, an estimate that is more than three times higher than existing estimates that attribute the death of infants to poor air quality for these countries. Therefore, air control and prevention of air pollution have become significant issues. In order to achieve this goal, obtaining real-time air pollution concentration is necessary [5]. Moreover, sensors have been used in a wide range of applications [6, 7], which collect extensive air quality data. For the increased attention of air pollution, many researchers take a significant focus on air pollution and there are many relevant research studies about air pollution. The main machine learning methods applied to air pollution are as follows: artificial neural network (ANN), ensemble learning, support vector machine (SVM), and other hybrid models [8]. However, these existing prediction machine learning methods of air quality lack analyzing the reasons for the change of air pollution concentration because most of the prediction methods take more focus on the model selection and ignore the reasons for changing. Furthermore, since the framework of recent deep learning is very flexible, the model may be deep and complex in order to fit...
the dataset. Therefore, overfitting problems may exist in a single deep neural network model when the number of weights in the deep neural network model is large. This paper analyzes the inherent relation of PM2.5 with other meteorological data (i.e., dew point, humidity, atmospheric pressure, temperature, wind direction, accumulated wind speed, precipitation, and accumulated precipitation), season data, and time stamp data. By analyzing the correlation between PM2.5 and other auxiliary data (an hour before), extracted air pollution characteristics are used to cluster the dataset and build a deep ensemble network (EN) model to predict PM2.5 concentration. The input of the model is the PM2.5 concentration and auxiliary data of the previous eight hours while the output is the PM2.5 concentration of the next hour. The adaptive moment estimation (Adam) algorithm is used to replace stochastic gradient descent (SGD) to update weights to get higher accuracy. For the validation of the proposed method, hourly PM2.5 concentration and meteorological data at 3 stations in Shanghai from 01 January 2010 to 31 December 2015 are collected. The mean absolute error (MAE) and mean absolute percentage error (MAPE) are used as accuracy metrics to compare the overall performance of each algorithm.

The contributions of this study are summarized as follows:

(i) This study proposes an ensemble model based on RNN, LSTM, and GRU to predict the PM2.5 concentration of the next hour
(ii) This study proposes a cluster method based on wind direction to improve prediction performance
(iii) Wind direction has been proved to be related to PM2.5 concentration because the wind can carry or take away PM2.5

The remainder of this paper is organized as follows. The literature reviews on air quality prediction in Section 2. Section 3 analyzes the inherent correlation between PM2.5 concentration and other auxiliary data and shows the data preprocess. Section 4 introduces the process of the Adam algorithm and the proposed EN. The experimental results are shown in Section 5. Section 6 is the conclusion including contributions and future work.

2. Literature Reviews

In this section, the disadvantages and advantages of existing machine learning for air quality prediction models are discussed.

2.1. Traditional Machine Learning Methods and Neural Networks with Simple Structure. Traditional machine learning methods and neural networks with simple structure were applied in PM2.5 prediction. In 2015, Lary et al. [9] proposed a model based on machine learning to estimate PM2.5 concentration. They collected the hourly PM2.5 data from 55 countries to verify the performance of the proposed model. Though the method got certain results, it could not predict future PM2.5 concentration. Hooyberghs et al. [10] proposed a method that combines ANN and big data to predict air quality. For the validation of the proposed model, the pollutants data, traffic data, and weather data were selected by smartphone sensors. The results showed that the ANN model is skilled in air pollution prediction. Moreover, some variations of ANN were proposed to predict air quality concentration. For example, the recurrent neural network (RNN) was used by Prakash et al. in 2011. This model was used to forecast 1 h ahead concentration and daily mean and daily maximum concentration of various pollutants, and the experimental results demonstrate the practicability of the method [11]. Besides, a hybrid model is also one of the variations of ANN. In 2011, Feng et al. [12] proposed a hybrid model that combines SVM with a back-propagation neural network (BPNN) to forecast ozone concentration. SVM was used to classify the data into its corresponding categories, and a genetic algorithm (GA) optimized BPNN was employed to build the prediction model. They collected the data including temperature, humidity, wind speed, and ultraviolet radiation from March 2009 to July 2009 to validate the accuracy of the proposed method, and the results showed that the model had a great prediction capability which could be used to predict the ozone concentration of Beijing. Considering the long-term dependencies and spatial correlations of air pollution, Li et al. [13] proposed an extended model of the long short-term memory (LSTM) network to extract the inherent features of air pollution and predict air quality. For the validation of the method, some models including the spatiotemporal deep learning (STDL) model, the time delay neural network (TDNN) model, the autoregressive moving average (ARMA) model, the support vector regression (SVR) model, and the traditional LSTM network [14] were used as the comparison algorithm and the results demonstrated the superiority of the proposed method.

2.2. Complex Deep Neural Networks. In recent years, deep learning has promoted the development of PM2.5 prediction. More and more complex deep networks are applied in this field to obtain better fitting results. In 2018, Huang and Kuo [15] analyzed the source pie of PM2.5 and proposed a deep neural network model that combines the convolution neural network (CNN) and LSTM network in 2018. CNN is a weight sharing network, which is good at capturing local features [16, 17]. This innovation of this method was introducing the convolution layer to extract spatial dependencies of PM2.5 and long short-term memory to extract temporal dependencies. The experimental results were compared with SVM, random forest (RD), decision tree (DT), NN, and LSTM algorithms and showed that PM2.5 concentration prediction models based on deep neural networks (e.g., NN, RNN, CNN, and LSTM) are better than the models based on traditional machine learning methods (e.g., SVM, RD, and DT). Considering the spatiotemporal dependence of PM2.5, Xie et al. [18] proposed a CGRU model based on CNN and gated recurrent unit (GRU) to predict PM2.5 concentration in the next six hours in 2019. CNN is used to extract spatial correlation features, and GRU further extracts long-term correlation features. Experimental results showed
that the proposed model was better than the traditional time series models (including LSTM, GRU, and ARIMA). GRU is a variant of LSTM, and the network structure is simpler than LSTM [19, 20]. In 2019, Tao et al. [21] proposed the CBGRU model based on 1D convnets and bidirectional GRU. On the basis of the bidirectional gated recurrent unit (BGRU), this method added the convolution layer and pool layer which can extract the PM2.5 features more easily. In 2020, Xay_souk et al. [22] developed two models including the LSTM model and the deep autoencoder (DAE) model to predict the particle concentration in the next hour. The experimental results showed that LSTM was better than DAE in predicting the particle concentration. In 2020, Kaya and Oguducu [23] proposed a new air quality prediction model based on deep learning, namely, deep flexible sequence. They used hourly data from Istanbul, Turkey, from 2014 to 2018 to predict air pollution before 4, 12, and 24 hours. This model is a hybrid and flexible deep model, which includes long short-term memory and convolutional neural network (CLSTM). On this basis, Li et al. [24] developed a deep CNN-LSTM method based on attention (ACLSTM), which includes the one-dimensional CNN, LSTM network, and attention network for urban PM2.5 concentration prediction. However, the attention layer is applied between the hidden layer and the output layer, which cannot explain the correlation between predictors and pollutants. Considering there are rare monitoring stations in a vast area, Ma et al. [25] proposed a deep spatiotemporal prediction method based on bidirectional LSTM and inverse distance weighting which can predict the PM2.5 concentration in the area without monitoring stations. Qi et al. [26] proposed a deep air learning method which provided novel ideas of interpolation, prediction, and feature analysis.

2.3. Ensemble Neural Networks. However, complex and deep network structure causes the decline of generalization ability, which leads to bad performance in other datasets. Hornik et al. [27] had proved that a single-layer ANN could approach the function with any complexity. However, how to make an appropriate network configuration was an NP-hard problem which influenced the generalization ability of the network. For solving the problem, Hansen and Salamon [28] proposed an ensemble neural network to provide a simple and feasible method. By this method, each of NN in the system was trained separately and the predictions of NN were synthesized as the final results.

3. Data Analysis

This paper collected hourly PM2.5 concentration and meteorological data at 3 stations in Shanghai from 01 January 2010 and 31 December 2015 from the UCI database [29]. The Pearson correlation coefficient is used to analyze the inherent correlation between PM2.5 and other auxiliary data of an hour before. The extracted features are used to choose the appropriate activation functions and train the EN which combined the RNN, LSTM, and GRU network to predict PM2.5 concentration. Before training the models, data preprocessing is necessary. This section analyzes the inherent correlation of PM2.5 and other auxiliary data in Section 3.1, and the data preprocessing is illustrated in Section 3.2.

3.1. Analyzing the Inherent Correlation of PM2.5 and Other Auxiliary Data. First of all, this study analyzes the spatial-temporal characteristics of three monitoring stations in Shanghai. They are located in Jingan, American consulate, and Xuhui separately as is shown in Figure 1. The autocorrelation function which is shown as Equation (1) is applied to measure the temporal correlation of each station, and more details are given in Reference [30]. In Table 1, the correlation values of three stations between PM2.5 concentration of an hour before and PM2.5 concentration of an hour later are above 0.95, which demonstrates that PM2.5 has a strong correlation in time.

\[ R(\tau) = \frac{E[(X_t - \mu)(X_{t+\tau} - \mu)]}{\sigma^2} , \]  

where \( X \) denotes the PM2.5 concentration of each hour, \( \mu \) is the expectation of \( X \), \( \tau \) is the time delay, \( \sigma \) is the standard deviation, and \( E \) is the expectation function. Existing studies have proved that meteorological factors play a significant role in air pollutant concentration [31, 32]. Therefore, it is necessary to find out the relationship between meteorological factors and PM2.5 concentration. The Pearson correlation coefficients of PM2.5 and other auxiliary data are shown in Table 1. The PM2.5 concentration data is the next hour data, and the auxiliary data is the before hour data. DEWP stands for dew point, HUMI stands for humidity, PRES stands for atmospheric pressure, TEMP stands for temperature, CV stands for no wind, NE stands for northeast wind, SE stands for southeast wind, SW stands for southwest wind, NW stands for northwest wind, lws stands for accumulated wind speed, and lprec stands for accumulated precipitation. As is shown in Table 1, except for HUMI, precipitation, lprec, and spring, whose values are below 0.10, other auxiliary data values are above 0.10. In the respect of season, summer and autumn show a negative correlation with PM2.5 data concentration while winter has a positive correlation with PM2.5 data concentration. In the respect of wind direction, there is a negative correlation between PM2.5 data concentration and east wind while west wind has a positive correlation with PM2.5 data concentration. The Pearson correlation coefficient functions are shown as Equation (2), and more details are given in Reference [30].

\[ R(X, Y) = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} , \]  

where \( \sigma_X \) is the standard deviation of \( X \) and \( \sigma_Y \) is the standard deviation of \( Y \). \( \mu_X \) is the expectation of \( X \), and \( \mu_Y \) is the expectation of \( Y \).

For intuitively observing the correlation between PM2.5 concentration and meteorological data, the violin plots are shown in Figure 2. The abscissa is the interval of meteorological data, and the ordinate stands for the PM2.5 concentration in the interval (e.g., the temperature value ranges from

\[ \text{Temperature Range} \]
-3 to 41, which is divided into 5 intervals as abscissa). The wider the image is, the more the number of data is. The division of interval varies with meteorological data. For the space limitation, this part only lists the violin plots of Jingan. Based on the correlation coefficient between wind directions and PM2.5 concentration, the dataset is divided into two parts: (1) west wind and no wind and (2) east wind. Two datasets are used to train and test the NN model. Six groups of controlled experiments are set to demonstrate the performance of the proposed method. Each group generates two models: (1) the NN model with the full dataset and (2) the NN model with the cluster method. In these groups, the number of dense layers of NN is 2 and the number of neurons is set from a candidate set of {5, 10, 15, 20, 25, 30}. Table 2 shows the experimental results for each test running. NN+CLU denotes NN based on the cluster method. In all groups, NN+CLU has the best performance (i.e., NN+CLU: 6.56 of MAE and 17.77% of MAPE in Group 1; NN+CLU: 6.87 of MAE and 19.65% of MAPE in Group 2; NN+CLU: 7.12 of MAE and 19.78% of MAPE in Group 3; NN+CLU: 6.92 of MAE and 19.11% of MAPE in Group 4; NN+CLU: 6.83 of MAE and 19.00% of MAPE in Group 5; and NN+CLU: 7.15 of MAE and 20.14% of MAPE in Group 6) which shows that the accuracy can be improved by the cluster method based on wind directions. When it is west direction, the PM2.5 concentration is higher, for Shanghai is located in the eastern coastal area of China, whose west is the inland. The west wind carries inland pollution, and no wind is not conducive to air circulation. On the contrary, the PM2.5 concentration is smaller, for the east wind carries the air from the ocean. The results illustrate that the cluster method based on wind directions can extract the data features effectively and get better predicted results.

3.2 Data Preprocessing. Hourly PM2.5 concentration, season data, and meteorological data at 3 stations in Shanghai from 01 January 2010 to 31 December 2015 are collected to test the performance of the proposed hourly PM2.5 concentration and meteorological data at 3 stations in Shanghai from 01 January 2010 to 31 December 2015 which are collected to test the performance of the proposed method. In the data preprocessing stage, the records with abnormal values or missing values are deleted firstly. Secondly, the wind direction data are changed into binary codes to enhance the prediction performance. Wind direction index data have 5 unique categorical values, and each wind direction index is transferred to a 5-dimensional vector (e.g., northwest is assigned as [0, 0, 0, 1]). Similarly, season index data have 4 unique categorical values, and each season index is transferred to a 4-dimensional vector (e.g., spring is assigned as [1, 0, 0, 0]). Thirdly, data is processed by the min-max normalization method and compressed from 0 to 1 for a better training effect. The formula is shown as Equation (3), and more details are given in Reference [33].

$$z_i = \frac{x_i - \min_{1 \leq n \leq N} (x_n)}{\max_{1 \leq n \leq N} (x_n) - \min_{1 \leq n \leq N} (x_n)}, \quad (3)$$

where $n$ denotes $n$-th records and $N$ is the number of records. $z$ is the normalized data ranging from 0 to 1.

4. The Proposed Ensemble Network Model for Prediction with Adam

The proposed EN model can be divided into three submodels: RNN, LSTM, and GRU. Three network models are used...
|                  | Jingan | US_Post | Xuhui | DEWP  | HUMI  | PRES  | TEMP  | lws   | Precipitation | Iprec | CV    | NE    | SE    | SW    | NW    | Spring | Summer | Autumn | Winter |
|------------------|--------|---------|-------|-------|-------|-------|-------|-------|---------------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|
| Jingan           | 0.97   | 0.96    | 0.94  | -0.22 | -0.06 | 0.14  | -0.21 | -0.23 | -0.07         | -0.09 | 0.12  | -0.23 | -0.14 | 0.15  | 0.27  | 0.02   | -0.15  | -0.15  | 0.29   |
| US_Post          | 0.96   | 0.96    | 0.91  | -0.33 | -0.08 | 0.25  | -0.33 | -0.22 | -0.08         | -0.09 | 0.11  | -0.19 | -0.16 | 0.12  | 0.28  | 0.03   | -0.25  | -0.15  | 0.36   |
| Xuhui            | 0.94   | 0.91    | 0.97  | -0.25 | -0.08 | 0.17  | -0.24 | -0.23 | -0.08         | -0.09 | 0.12  | -0.23 | -0.13 | 0.15  | 0.26  | 0.00   | -0.16  | -0.14  | 0.31   |
| DEWP             | -0.22  | -0.33   | -0.25 | 0.99  | 0.42  | -0.85 | 0.87  | -0.01 | 0.09          | 0.07  | -0.06 | 0.20  | 0.09  | -0.22 | 0.12  | 0.62   | 0.15   | -0.64  |        |
| HUMI             | -0.06  | -0.08   | -0.08 | 0.42  | 0.95  | -0.21 | -0.05 | 0.03  | 0.15          | 0.15  | 0.04  | 0.08  | 0.07  | -0.15 | -0.07 | 0.09   | 0.13   | 0.01   | -0.05  |
| PRES             | 0.14   | 0.25    | 0.17  | -0.85 | -0.21 | 1.00  | -0.84 | 0.00  | -0.09         | -0.07 | 0.07  | 0.17  | -0.20 | -0.18 | 0.15  | -0.07  | -0.67  | 0.11   | 0.61   |
| TEMP             | -0.21  | -0.33   | -0.24 | 0.87  | -0.05 | -0.84 | 0.99  | -0.03 | 0.02          | 0.01  | -0.08 | 0.11  | 0.18  | 0.18  | -0.20 | -0.07  | 0.61   | 0.15   | -0.68  |
| lws              | -0.23  | -0.22   | -0.23 | -0.01 | 0.03  | 0.00  | -0.03 | 0.95  | 0.03          | 0.06  | -0.13 | 0.12  | 0.04  | -0.15 | -0.01 | 0.03   | -0.04  | 0.03   | -0.02  |
| Precipitation    | -0.07  | -0.08   | -0.08 | 0.09  | 0.15  | -0.09 | 0.02  | 0.03  | 0.38          | 0.21  | -0.02 | 0.03  | 0.00  | -0.02 | -0.02 | 0.00   | 0.05   | -0.01  | -0.03  |
| Iprec            | -0.09  | -0.09   | -0.09 | 0.07  | 0.15  | -0.07 | 0.01  | 0.06  | 0.21          | 0.91  | -0.02 | 0.07  | -0.03 | -0.03 | -0.02 | 0.02   | 0.02   | -0.02  |        |
| CV               | 0.12   | 0.11    | 0.12  | -0.06 | 0.04  | 0.07  | -0.08 | -0.13 | -0.02         | -0.02 | 0.31  | -0.07 | -0.06 | 0.05  | -0.03 | -0.08  | -0.09  | 0.15   | 0.01   |
| NE               | -0.23  | -0.19   | -0.23 | -0.05 | 0.08  | 0.17  | -0.11 | 0.12  | 0.03          | 0.07  | -0.07 | 0.76  | -0.38 | -0.27 | -0.23 | 0.15   | 0.09   | -0.11  | -0.15  |
| SE               | -0.14  | -0.16   | -0.13 | 0.20  | 0.07  | -0.20 | 0.18  | 0.04  | 0.00          | -0.03 | -0.06 | 0.38  | 0.75  | -0.13 | -0.30 | 0.02   | 0.14   | -0.11  | -0.05  |
| SW               | 0.15   | 0.12    | 0.15  | 0.09  | -0.15 | -0.18 | 0.18  | -0.15 | -0.02         | -0.03 | 0.05  | -0.27 | -0.13 | 0.62  | -0.10 | -0.09  | -0.11  | 0.03   | 0.17   |
| NW               | 0.27   | 0.28    | 0.26  | -0.22 | -0.07 | 0.15  | -0.20 | -0.01 | -0.02         | -0.02 | -0.03 | -0.23 | -0.30 | -0.10 | 0.75  | 1.00   | -0.33  | -0.36  | -0.35  |
to predict PM2.5 concentration, respectively, and the results of them are combined by the weighted average method. The weights of EN are optimized by the Adam algorithm. Section 4.1 illustrates the EN, and Section 4.2 introduces the algorithm process of Adam.

4.1. Ensemble Network. As shown in Figure 3, the proposed EN model consisting of the RNN model, LSTM model, and GRU model can be applied to predict PM2.5 concentration. The establishment of the proposed model can be divided into two sections: training independent network model and combining the results of every network model. This section introduces the training stage of the NN, RNN, LSTM, and GRU models in Sections 4.1.1 and 4.1.2. The combining stage is illustrated in Section 4.1.3.

4.1.1. Neural Network. In this stage, historical and auxiliary data are used to train each network model in the EN model separately. The NN model is employed to analyze the interaction effects of input parameters and get the prediction. It is divided into three layers: input layer, hidden layer, and output layer. Besides, the rectified linear unit (RELU) function is applied as an activation function which is added behind the output layer to produce a nonlinear prediction. The
Adam an algorithm for stochastic optimization is used to optimize the weights instead of SGD.

For the training of NNs, the PM2.5 concentration data and auxiliary data of the previous hour in three stations are used as the inputs of NNs while the PM2.5 concentration data of the next hour in three stations are used as the outputs of NNs. The full connected layers are used as the hidden layer to analyze the inherent correlation of parameters. The historical data are applied to train the models, and the weights of models are optimized by the Adam algorithm.

4.1.2. Recurrent Neural Network. Besides NN models, other deep learning methods are applied to prediction problems (e.g., RNN models, LSTM models). The inputs of the RNN, LSTM, and GRU models are different from NN models for each neuron of the NN’s input layers is a single sequence element while each neuron of the input layer in RNN, LSTM, and GRU is a vector which is encoded by the past sequence elements. As has been said before, PM2.5 concentration data have a strong correlation in time. RNN, LSTM, and GRU are applied to predict PM2.5 concentration for they are experts in dealing with time series problems compared with NN models. LSTM an extended model of RNN differs from RNN in learning long-time dependence for there is a phenomenon of gradient disappearance in RNN. GRU an extended model of LSTM differs from LSTM in internal structure for LSTM has three gates and GRU has only two. In the training stage of the RNN, LSTM, and GRU models, eight hours of PM2.5 concentration data along with meteorological data is regarded as input which is different from the NN model.

4.1.3. Combining State of the Ensemble Network. In Figure 3, each submodel in EN can predict the PM2.5 concentration independently and all the results will be integrated to produce the final prediction results. In this research, the number of hidden layer neurons ranged from 5 to 30 increasing by 5 at a time. The weighted average method is applied to integrate all the prediction results of submodels. In order to obtain the weight of each submodel, 10% of the dataset is selected as the validation set. The accuracy of each submodel is applied in Softmax to get weight. The Softmax formula is shown in Equation (4), and more details are given in [34].

\[
\sum_{i=1}^{n} e^{z_i}
\]

where \(z_1, z_2, \ldots, z_i\) denote the accuracy of submodels on the validation set, \(n\) is the number of submodels, and \(w_1, w_2, \ldots, w_i\) are the weight of submodels. \(e\) denotes the natural exponential. The final result of the proposed model can be computed as

\[
\text{accuracy} = \sum_{i=1}^{n} w_i \cdot z_i.
\]

This section sets six groups of controlled experiments. Each group generates seven models consisting of NN, RNN, LSTM, EN, EN1, EN2, EN3, EN4, and EN5, where EN denotes a combination of RNN, LSTM, and GRU; EN1 denotes a combination of NN and RNN; EN2 denotes a combination of NN and LSTM; EN3 denotes a combination of RNN and LSTM; EN4 denotes a combination of NN, RNN, and LSTM; EN5 denotes a combination of NN, RNN, LSTM, and GRU. In these groups, the number of dense layers of NN, LSTM, and GRU is 1 and the number of neurons is generated from a candidate set of {5, 10, 15, 20, 25, 30}. For the comparisons of the overall performance of different optimized algorithms, MAE and MAPE are used as accuracy metrics for the experiments.

Table 3 shows the experimental results for each test running. In all groups, EN (including EN, EN1, EN2, EN3, EN4, and EN5) has the best performance (i.e., EN1: 6.51 of MAE and 19.35% of MAPE in Group 1; EN: 6.35 of MAE and 17.36% of MAPE in Group 2; EN: 6.19 of MAE and 16.20% of MAPE in Group 3; EN: 6.20 of MAE and 16.28% of MAPE in Group 4; EN2: 6.71 of MAE and 19.56% of MAPE in Group 5; and EN3: 6.47 of MAE and 17.80% of MAPE in Group 6). The results illustrate that the combination of different models is effective to predict PM2.5 concentration and decreases the error of overfitting.

4.2. Adam Optimization. This paper adopted Adam [35] an optimization algorithm to replace the traditional SGD. It can update the weights of the neural network iteratively based on the dataset. The Adam optimization algorithm is an extension of the SGD algorithm, which is widely used in deep learning applications recently. The Adam algorithm is different from traditional SGD for the latter keeps a single learning rate to update all weights and the learning rate does not change in the training process while the former designs...
an independent adaptive learning rate for different parameters. The Adam algorithm will be introduced in detail next.

Assume that \( f \) is the objective function and \( \theta \) are the parameters which require to be optimized. \( g_t \) stands for the gradient which can be expressed as Equation (6), and more details of formulas of this section are given in [35].

\[
g_t = \nabla \theta f_t(\theta_{t-1}), \tag{6}
\]

where the \( f_1(\theta), f_2(\theta), \ldots, f_t(\theta) \) stand for the function values of time step 1 to \( t \). \( m_t \) and \( v_t \) stand for the exponential moving averages of the gradient (i.e., biased first moment estimate) and the squared gradient (i.e., biased second raw moment estimate) which are employed to update weights separately and their formulas are shown as

\[
m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t, \tag{7}
\]

\[
v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2, \tag{8}
\]

where \( \beta_1 \) and \( \beta_2 \) ranging from 0 to 1 control the exponential decay rates for the moment estimates.

In the beginning, \( m_t \) and \( v_t \) are near 0, for they are set as 0 and the decay rate is close to 1. For the sake of counteracting the bias at the beginning, bias-corrected estimates \( \hat{m}_t \) and \( \hat{v}_t \) are introduced as

\[
\hat{m}_t = \frac{m_t}{1 - (1 - \beta_1)^t}, \quad \hat{v}_t = \frac{v_t}{1 - (1 - \beta_2)^t},
\]

as shown in Equation (9), where \( \beta_1 \) and \( \beta_2 \) ranging from 0 to 1 control the exponential decay rates for the moment estimates.

In the beginning, \( m_t \) and \( v_t \) are near 0, for they are set as 0 and the decay rate is close to 1. For the sake of counteracting the bias at the beginning, bias-corrected estimates \( \hat{m}_t \) and \( \hat{v}_t \) are introduced as

\[
\hat{m}_t = \frac{m_t}{1 - (1 - \beta_1)^t}, \quad \hat{v}_t = \frac{v_t}{1 - (1 - \beta_2)^t}, \tag{9}
\]

In the beginning, \( m_t \) and \( v_t \) are near 0, for they are set as 0 and the decay rate is close to 1. For the sake of counteracting the bias at the beginning, bias-corrected estimates \( \hat{m}_t \) and \( \hat{v}_t \) are introduced as

\[
\hat{m}_t = \frac{m_t}{1 - (1 - \beta_1)^t}, \quad \hat{v}_t = \frac{v_t}{1 - (1 - \beta_2)^t}, \tag{9}
\]

In the beginning, \( m_t \) and \( v_t \) are near 0, for they are set as 0 and the decay rate is close to 1. For the sake of counteracting the bias at the beginning, bias-corrected estimates \( \hat{m}_t \) and \( \hat{v}_t \) are introduced as

\[
\hat{m}_t = \frac{m_t}{1 - (1 - \beta_1)^t}, \quad \hat{v}_t = \frac{v_t}{1 - (1 - \beta_2)^t}, \tag{9}
\]

In the beginning, \( m_t \) and \( v_t \) are near 0, for they are set as 0 and the decay rate is close to 1. For the sake of counteracting the bias at the beginning, bias-corrected estimates \( \hat{m}_t \) and \( \hat{v}_t \) are introduced as

\[
\hat{m}_t = \frac{m_t}{1 - (1 - \beta_1)^t}, \quad \hat{v}_t = \frac{v_t}{1 - (1 - \beta_2)^t}, \tag{9}
\]

In the beginning, \( m_t \) and \( v_t \) are near 0, for they are set as 0 and the decay rate is close to 1. For the sake of counteracting the bias at the beginning, bias-corrected estimates \( \hat{m}_t \) and \( \hat{v}_t \) are introduced as

\[
\hat{m}_t = \frac{m_t}{1 - (1 - \beta_1)^t}, \quad \hat{v}_t = \frac{v_t}{1 - (1 - \beta_2)^t}, \tag{9}
\]

In the beginning, \( m_t \) and \( v_t \) are near 0, for they are set as 0 and the decay rate is close to 1. For the sake of counteracting the bias at the beginning, bias-corrected estimates \( \hat{m}_t \) and \( \hat{v}_t \) are introduced as

\[
\hat{m}_t = \frac{m_t}{1 - (1 - \beta_1)^t}, \quad \hat{v}_t = \frac{v_t}{1 - (1 - \beta_2)^t}, \tag{9}
\]

In the beginning, \( m_t \) and \( v_t \) are near 0, for they are set as 0 and the decay rate is close to 1. For the sake of counteracting the bias at the beginning, bias-corrected estimates \( \hat{m}_t \) and \( \hat{v}_t \) are introduced as

\[
\hat{m}_t = \frac{m_t}{1 - (1 - \beta_1)^t}, \quad \hat{v}_t = \frac{v_t}{1 - (1 - \beta_2)^t}, \tag{9}
\]

In the beginning, \( m_t \) and \( v_t \) are near 0, for they are set as 0 and the decay rate is close to 1. For the sake of counteracting the bias at the beginning, bias-corrected estimates \( \hat{m}_t \) and \( \hat{v}_t \) are introduced as

\[
\hat{m}_t = \frac{m_t}{1 - (1 - \beta_1)^t}, \quad \hat{v}_t = \frac{v_t}{1 - (1 - \beta_2)^t}, \tag{9}
\]
Table 3: The accuracy comparisons between NN, RNN, LSTM, and EN with different numbers of neurons.

| Model                  | The number of neurons of dense layers | MAE  | MAPE  |
|------------------------|--------------------------------------|------|-------|
| Group 1                |                                       |      |       |
| NN                     | 5                                    | 6.99 | 19.59%|
| RNN                    | 5                                    | 7.23 | 20.17%|
| LSTM                   | 5                                    | 6.63 | 20.01%|
| GRU                    | 5                                    | 6.73 | 20.23%|
| EN1 (NN+RNN)           | [5, 5]                               | 7.19 | 20.31%|
| EN2 (NN+LSTM)          | [5, 5]                               | 6.62 | 19.65%|
| EN3 (RNN+LSTM)         | [5, 5]                               | 6.59 | 19.35%|
| EN4 (NN+RNN+LSTM)      | [5, 5, 5]                            | 6.73 | 19.61%|
| EN5 (NN+RNN+LSTM+GRU)  | [5, 5, 5, 5]                         | 6.72 | 19.60%|
| EN (RNN+LSTM+GRU)      | [5, 5]                               | 6.60 | 19.99%|
| Group 2                |                                       |      |       |
| EN1 (NN+RNN)           | [10, 10]                             | 6.54 | 18.30%|
| EN2 (NN+LSTM)          | [10, 10]                             | 7.02 | 20.76%|
| EN3 (RNN+LSTM)         | [10, 10]                             | 6.84 | 20.68%|
| EN4 (NN+RNN+LSTM)      | [10, 10, 10]                         | 6.65 | 19.49%|
| EN5 (NN+RNN+LSTM+GRU)  | [10, 10, 10, 10]                     | 6.67 | 19.23%|
| EN (RNN+LSTM+GRU)      | [10, 10]                             | 6.35 | 17.36%|
| Group 3                |                                       |      |       |
| EN1 (NN+RNN)           | [15, 15]                             | 6.42 | 17.68%|
| EN2 (NN+LSTM)          | [15, 15]                             | 6.49 | 17.06%|
| EN3 (RNN+LSTM)         | [15, 15]                             | 6.22 | 16.25%|
| EN4 (NN+RNN+LSTM)      | [15, 15, 15]                         | 6.26 | 16.64%|
| EN5 (NN+RNN+LSTM+GRU)  | [15, 15, 15, 15]                     | 6.25 | 16.56%|
| EN (RNN+LSTM+GRU)      | [15, 15]                             | 6.19 | 16.20%|
| Group 4                |                                       |      |       |
| EN1 (NN+RNN)           | [20, 20]                             | 6.57 | 17.39%|
| EN2 (NN+LSTM)          | [20, 20]                             | 6.20 | 16.42%|
| EN3 (RNN+LSTM)         | [20, 20]                             | 6.46 | 16.57%|
| EN4 (NN+RNN+LSTM)      | [20, 20, 20]                         | 6.24 | 16.28%|
| EN5 (NN+RNN+LSTM+GRU)  | [20, 20, 20, 20]                     | 6.20 | 16.56%|
| EN (RNN+LSTM+GRU)      | [20, 20]                             | 6.28 | 16.46%|
| Group 5                |                                       |      |       |
| EN1 (NN+RNN)           | [25, 25]                             | 7.10 | 20.87%|
| EN2 (NN+LSTM)          | [25, 25]                             | 6.71 | 19.56%|
| EN3 (RNN+LSTM)         | [25, 25]                             | 7.13 | 21.57%|
| EN4 (NN+RNN+LSTM)      | [25, 25, 25]                         | 6.86 | 20.37%|
are introduced. The calculation formulas are shown as

\[ m_i = \frac{m_i}{1 - \beta_1^i}, \quad (9) \]

\[ v_i = \frac{v_i}{1 - \beta_2^i}, \quad (10) \]

The final updated formula of parameters is shown as

\[ \theta_i = \theta_{i-1} - \alpha \cdot \frac{m_i}{\sqrt{v_i + \epsilon}}, \quad (11) \]

where \( \alpha \) denotes the stepsize (i.e., learning rate) and is initialized to 0.001 (i.e., default value).

This section sets four groups of controlled experiments, and NN, RNN, LSTM, and GRU are applied to predict the PM2.5 concentration by using SGD, Adam, and Nadam separately. All the number of neurons of dense layers in these models is set to 15, and the performance of each algorithm in different networks is shown in Table 4. Compared with Adam and Nadam, SGD has the worst performance because it has a fixed learning rate leading to find global optimum difficultly. Except for the MAE of NN, Adam has better performance than Nadam [36] which proves that the Adam algorithm is effective in optimizing the PM2.5 predicting model.

In order to determine the values of \( \beta_1 \) and \( \beta_2 \), this section sets three groups of controlled experiments and EN (RNN, LSTM, and GRU) is employed to predict PM2.5 concentration with different \( \beta_1 \) and \( \beta_2 \). All the number of neurons of dense layers in these models is set to 15; the performance of different values of \( \beta_1 \) and \( \beta_2 \) is shown in Table 5. The default values of \( \beta_1 \) and \( \beta_2 \) (i.e., \( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \)) get the best performance comparing with other values (i.e., \( \beta_1 = 0.8, \beta_2 = 0.888 \) and \( \beta_1 = 0.7, \beta_2 = 0.777 \)).

5. Experimental Results and Analysis

This section uses hourly PM2.5 concentration and meteorological data at 3 stations in Shanghai from 01 January 2010 to 31 December 2015 to evaluate the proposed model. All the models including NN, RNN, LSTM, GRU, BGRU, CGRU, CBGRU, CLSTM, ACLSTM, and EN are trained on the Keras framework with TensorFlow backend. The learning rate is set to 0.001, and the epochs are set to 200. RELU is applied as an activation function for each layer of the network, and Adam is used as the optimized algorithm to optimize the weights. For the validation of the proposed method, MAE and MAPE are used as accuracy metrics to compare the overall performance of each model. MAE is an absolute value, and MAPE is a percentage, smaller values of which indicate better performance. Two metrics are given in Equations (12) and (13), respectively, and more details are given in Reference [30].

\[ \text{MAE} = \frac{1}{N} \sum_{n=1}^{N} |o_n - p_n|, \quad (12) \]

\[ \text{MAPE} = \frac{1}{N} \sum_{n=1}^{N} \frac{|o_n - p_n|}{o_n}, \quad (13) \]

where \( o_n \) is the value of the \( n \)-th observed data and \( p_n \) denotes the predicted value of the \( n \)-th predicted data. The values of two metrics (MAE, MAPE) are calculated for proposed EN as well as for NN, RNN, LSTM, GRU, BGRU, CGRU, CBGRU, CLSTM, and ACLSTM. Table 6 illustrates the detailed MAE and MAPE values of each algorithm. In the ranking of MAE from high to low, we have CBGRU (8.58%), CGRU (8.56%), RNN (7.23%), BGRU (7.22), NN (6.95), CLSTM (6.90), ACLSTM (6.86), GRU (6.56), LSTM (6.48), and EN (6.19). In the ranking of MAPE from high to low, we have CBGRU (24.73%), CGRU (20.50%), GRU (19.84%), NN (19.19%), BGRU (18.88%), RNN (18.49%), CLSTM (17.41%), LSTM (16.79%), ACLSTM (16.60%), and EN (16.20%). No matter MAE or MAPE, EN has the minimum values which confirms the advantages of EN. Compared with the traditional NN and RNN, LSTM and GRU have better performance in predicting PM2.5 which indicates that the PM2.5 concentration has a long-term dependence. As sub-models of the ensemble model, NN, RNN, LSTM, and GRU
ensemble network is a model that can adapt to different data structures, they may lead to overfitting in other datasets, which decreases the generalization ability of the models. The performance in their respective datasets with deep and complex structures, they may lead to overfitting in other datasets, which decreases the generalization ability of the models. The ensemble network is a model that can adapt to different datasets for the ensemble model consists of three submodels and the weight of each model depends on the prediction accuracy of the model in the dataset employed, which ensures the stability of the model in different datasets. Figure 4 shows the comparison between observed values and predicted values of the EN model. From the overall trend, the prediction data can better fit the observation data. This also proves that the proposed model can effectively predict the PM2.5 concentration in the next hour.

### 6. Conclusions and Future Work

Because of the flexibility of the network framework, many complex deep learning networks have been developed for air quality prediction. As far as we know, there is no uniform dataset in current air quality prediction research. Researchers collected datasets from different regions to train the network. Although these complex deep networks can well fit the data they use, they lack generalization ability. Therefore, this paper proposes an EN model to predict air pollution concentration by historical PM2.5 concentration, meteorological, and time stamp data. Considering that the submodel including RNN, LSTM, and GRU has quite good performance, each submodel of the EN model is trained, respectively, to get the accuracy and obtain the final model by a weighted average method. The weights of submodels are flexible because they are obtained by the accuracy of the validation set so that they can perform stably in different datasets. In addition, the ensemble of different networks is less involved in this field. As far as we know, Adam is adopted to optimize weights instead of SGD for it can adjust the learning rate adaptively to get an efficient training effect. A case study of the prediction of PM2.5 concentration in Shanghai of the People’s Republic of China is given in this research, and the dataset is divided into three parts: training data, validation data, and testing data. Training data are used to train the submodels of EN separately, validation data are applied to obtain the weight of each submodel, and testing data are adopted to compute MAE and MAPE for performance evaluation. The experimental evaluation is performed for EN, as well as other algorithms including NN, RNN, LSTM, GRU, BGRU, CBGRU, CLSTM, and ACLSTM. The experimental results demonstrate that the proposed method has the best performance which outperforms other algorithms. Several findings of this paper are as follows:

(i) Compared with the single model, the EN model has better generalization ability and predictive ability as validated by MAE and MAPE.

(ii) Wind direction has a significant impact on PM2.5 concentration for wind can carry or take away PM2.5.

(iii) Compared with SGD, the Adam algorithm avoid the local optimum effectively.

For the extension of this study, the prediction performance can be enhanced by adding human activities because it is one of the main reasons for environmental deterioration.

### Table 4: The accuracy comparisons of different models with different optimized algorithms.

| Model  | The number of neurons of dense layers | Algorithm | MAE  | MAPE  |
|--------|--------------------------------------|-----------|------|-------|
| NN     | 15                                   | SGD       | 7.13 | 22.38%|
| RNN    | 15                                   | Adam      | 7.11 | 19.89%|
| LSTM   | 15                                   | Nadam     | 6.75 | 20.11%|
| LSTM   | 15                                   | SGD       | 12.33| 32.45%|
| GRU    | 15                                   | Adam      | 6.57 | 16.77%|
| GRU    | 15                                   | Nadam     | 6.58 | 17.29%|
| GRU    | 15                                   | SGD       | 11.18| 30.38%|
| LSTM   | 15                                   | Adam      | 6.45 | 17.34%|
| LSTM   | 15                                   | Nadam     | 6.67 | 17.47%|
| LSTM   | 15                                   | SGD       | 9.19 | 25.00%|
| LSTM   | 15                                   | Adam      | 6.56 | 19.84%|
| LSTM   | 15                                   | Nadam     | 6.70 | 19.98%|

### Table 5: The accuracy comparisons of different values of $\beta_1$ and $\beta_2$ in the EN model.

| Values of $\beta_1$ and $\beta_2$ | The number of neurons of dense layers | MAE  | MAPE  |
|-----------------------------------|--------------------------------------|------|-------|
| $\beta_1 = 0.9$                  | 15                                   | 6.19 | 16.20%|
| $\beta_1 = 0.999$                | 15                                   | 6.26 | 16.44%|
| $\beta_1 = 0.8$                  | 15                                   | 6.26 | 16.44%|
| $\beta_1 = 0.888$                | 15                                   | 6.20 | 16.35%|
| $\beta_1 = 0.8$                  | 15                                   | 6.20 | 16.35%|
| $\beta_1 = 0.777$                | 15                                   | 6.20 | 16.35%|

### Table 6: The accuracy comparisons of different algorithms.

| Model    | MAE  | MAPE  |
|----------|------|-------|
| NN       | 6.95 | 19.19%|
| RNN      | 7.23 | 18.49%|
| LSTM     | 6.48 | 16.79%|
| GRU      | 6.56 | 19.84%|
| BGRU     | 7.22 | 18.88%|
| CBGRU    | 8.56 | 20.50%|
| CBGRU    | 8.58 | 24.73%|
| CLSTM    | 6.90 | 17.41%|
| ACLSTM   | 6.86 | 16.60%|
| EN (RNN+LSTM+GRU) | 6.19 | 16.20%|
especially in holidays. Furthermore, this paper found that wind direction has a significant influence on PM2.5 concentration because the wind will bring or take away PM2.5. However, it is uncertain that it can do in the areas with high mountains. Therefore, embedding the influence of topographical factors can become a research direction in the future. However, limited by lacking human activity and topographical data, this paper only analyzes the impact of meteorological data on PM2.5.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Nos. 61906043, 61877010, 11501114, and 11901100), Fujian Natural Science Funds (No. 2019J01243), Funds of Department of Education, Fujian Province (No. JAT190026), and Fuzhou University (Nos. 510730/XRC-18049, and 510730/XRC-18050).

References

[1] F. Launay, “7 million deaths annually linked to air pollution,” Central European Journal of Public Health, vol. 22, no. 1, pp. 53–59, 2014.

[2] A. Kurt and A. B. Oktay, “Forecasting air pollutant indicator levels with geographic models 3days in advance using neural networks,” Expert Systems with Applications, vol. 37, no. 12, pp. 7986–7992, 2010.

[3] D. W. Dockery, C. A. Pope, X. Xu et al., “An association between air pollution and mortality in six U.S. cities,” New England Journal of Medicine, vol. 329, no. 24, pp. 1753–1759, 1993.

[4] S. Heft-Neal, J. Burney, E. Bendavid, and M. Burke, “Robust relationship between air quality and infant mortality in Africa,” Nature, vol. 559, no. 7713, pp. 254–258, 2018.

[5] Y. Zheng, F. Liu, and H. P. Hsieh, “U-air: when urban air quality inference meets big data,” in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1436–1444, Chicago, USA, 2013.

[6] X. S. Xue and J. F. Chen, “Optimizing sensor ontology alignment through compact co-firefly algorithm,” Sensors, vol. 20, no. 7, article 2056, 2020.

[7] X. S. Xue and J. F. Chen, “Using compact evolutionary Tabu search algorithm for matching sensor ontologies,” Swarm and Evolutionary Computation, vol. 48, pp. 25–30, 2019.

[8] Y. Rybarczyk and R. Zalakeviciute, “Machine learning approaches for outdoor air quality Modelling: A Systematic Review,” Applied Sciences, vol. 8, no. 12, article 2570, 2018.

[9] D. J. Lary, T. Lary, and B. Sattler, “Using machine learning to estimate global PM2.5 for environmental health studies,” Environmental Health Insights, vol. 9, no. 51, article EHI. S15664, 2020.

[10] J. Hooyberghs, C. Mensink, G. Dumont, F. Fierens, and O. Brasseur, “A neural network forecast for daily average PM concentrations in Belgium,” Atmospheric Environment, vol. 39, no. 18, pp. 3279–3289, 2005.

[11] U. K. Prakash, K. Kumar, and V. K. Jain, “A wavelet-based neural network model to predict ambient air pollutants’ concentration,” Environmental Modeling & Assessment, vol. 16, no. 5, pp. 503–517, 2011.

[12] Y. Feng, W. Zhang, D. Sun, and L. Zhang, “Ozone concentration forecast method based on genetic algorithm optimized back propagation neural networks and support vector machine data classification,” Atmospheric Environment, vol. 45, no. 11, pp. 1979–1985, 2011.

[13] X. Li, L. Peng, X. Yao et al., “Long short-term memory neural network for air pollutant concentration predictions: method

Figure 4: PM2.5 concentration forecasting results of the EN model.
development and evaluation,” *Environmental Pollution*, vol. 231, Part 1, pp. 997–1004, 2017.

[14] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] C. J. Huang and P. H. Kuo, “A deep CNN-LSTM Model for particulate matter (PM2.5) forecasting in smart cities,” *Sensors*, vol. 18, no. 7, article 2220, 2018.

[16] I. S. Krizhevsky and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[17] Y. Kim, “Convolutional neural networks for sentence classification,” 2014, https://arxiv.org/abs/1408.5882.

[18] H. F. Xie, L. Ji, Q. Wang, and Z. J. Jia, “Research of PM2.5 prediction system based on CNNs-GRU in Wuxi urban area,” *Earth and Environmental Science*, vol. 300, 2020.

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” 2014, https://arxiv.org/abs/1412.3555.

[20] X. Luo, W. Zhou, W. Wang, Y. Zhu, and J. Deng, “Attention-based relation extraction with bidirectional gated recurrent unit and highway network in the analysis of geological data,” *IEEE Access*, vol. 6, pp. 5705–5715, 2017.

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

[22] T. Xayasouk, H. Lee, and G. Lee, “Air pollution prediction using long short-term memory (LSTM) and deep autoencoder (DAE) models,” *Sustainability*, vol. 12, no. 6, p. 2570, 2020.

[23] K. Kaya and S. G. Oguducu, “Deep flexible sequential (DFS) model for air pollution forecasting,” *Scientific Reports*, vol. 10, no. 1, 2020.

[24] S. Z. Li, G. Xie, J. C. Ren, L. Guo, L. Y. Y. Yang, and X. Y. Xu, “Urban PM2.5 concentration prediction via attention-based CNN-LSTM,” *Applied Sciences*, vol. 10, no. 6, 2020.

[25] J. Ma, Y. X. Ding, V. J. L. Gan, C. Q. Lin, and Z. W. Wan, “Spatiotemporal prediction of PM2.5 concentrations at different time granularities using IDW-BLSTM,” *IEEE Access*, vol. 7, pp. 107897–107907, 2019.

[26] Z. G. Qi, T. C. Wang, G. J. Song, W. S. Hu, X. Li, and Z. F. Zhang, “Deep air learning: interpolation, prediction, and feature analysis of fine-grained air quality,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2285–2297, 2018.

[27] K. Hornik and M. Stinchcombe, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[28] L. K. Hansen and P. Salamon, “Neural network ensembles,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 12, no. 10, pp. 999–1001, 1990.

[29] S. X. Chen, “PM2.5 data of five Chinese cities data set,” 2017, https://archive.ics.uci.edu/ml/datasets/PM2.5+Data+of+Five+Chinese+Cities.

[30] S. Kounev, K. D. Lange, and J. V. Kistowski, “Review of basic probability and statistics,” in *Systems Benchmarking*, 2020.

[31] J. He, Y. Yu, Y. Xie et al., “Numerical model-based artificial neural network model and its application for quantifying impact factors of urban air quality,” *Water, Air, & Soil Pollution*, vol. 227, no. 7, article 235, 2016.

[32] Y. Bai, Y. Li, X. Wang, J. Xie, and C. Li, “Air pollutants concentrations forecasting using back propagation neural network based on wavelet decomposition with meteorological condi-