Environmental Science and Pollution Research
https://doi.org/10.1007/s11356-020-08948-1

RESEARCH ARTICLE

Combining forward with recurrent neural networks for hourly air quality prediction in Northwest of China

Zhili Zhao[1] · Jian Qin[1] · Zhaoshuang He[1] · Huan Li[1] · Yi Yang[1] · Ruisheng Zhang[1]

Received: 25 October 2019 / Accepted: 17 April 2020

Abstract

Data-driven statistical air quality prediction methods usually build models fast with moderate accuracy and have been studied a lot in recent years. However, due to the complexity of air quality prediction which usually involves multiple factors, such as meteorological, spatial, and temporal properties, it is still a challenge to propose a model with required accuracy. In this paper, we propose a hybrid ensemble model CERL to exploit the merits of both forward neural networks and recurrent neural networks that are designed for handling time serial data to predict air quality hourly. Measured air pollutant factors including Air Quality Index (AQI), $PM_{2.5}$, PM_{10}, CO, SO_2, NO_2, and O_3 are used as input to predict air quality from 1 to 8 h ahead. Based on the air quality prediction evaluation in Lanzhou and Xi'an, which are two important provincial capitals in Northwest China, CERL provides better performance over other baseline models. Moreover, as the step length increases, CERL has more obvious improvement. For example, the improvements of CERL in the 1-step, 3-step, 5-step, and 8-step prediction for $PM_{2.5}$ in Lanzhou are 1.82%, 8.01%, 9.98%, and 20.03%, respectively. The superiority of CERL is also proved by a hypothesis Diebold Mariano test with level of significance 5%.

Keywords Air quality · Machine learning · Serial data · SSA · Sliding window · Neural network

Introduction

Due to rapid population growth and backward economic levels, air pollution has been one of the major problems perplexing many developing countries. According to the latest world air quality report released by AirVisual (2019), Asian locations dominate the highest 100 average $PM_{2.5}$ levels during 2018, with cities in India, China, Pakistan, and Bangladesh occupying the top 50 cities (AirVisual 2018). China is the largest developing country in the world, and many cities of China have suffered from serious air pollution in the past few years, such as Hotan, Shijiazhuang, Baoding, Xianyang, Jiaozuo, and Cangzhou. Although China's air pollution exposures have stabilized and even begun to decline slightly after several years of strict restrictions on industrial emissions and the use of fossil fuels for indoor heating and cooking (HEI and IHME 2018), efforts are still needed to protect environment at a high level. Sulfur oxides, carbon oxides, nitrogen oxides, hydrocarbons, particulate matter 10 (PM_{10}), and particulate matter 2.5 ($PM_{2.5}$) in the atmosphere are the main contributors to air pollution, and many efforts have been put into predicting air quality based on the observations of scattered air monitoring stations.

Deterministic methods usually build simulation models to simulate and predict the diffusion and transport process of atmospheric pollutants (Ma et al. 2019). However, such methods suffer from large computation costs and low prediction accuracy if underlying atmospheric conditions are complex and involve a large amount of observed data. Moreover, it is necessary for IT technologists to know specific domain knowledge for parameter identification. Machine learning methods are another kind of approaches for air quality prediction based on a large amount of observed data. In recent years, researchers have employed many machine learning methods to predict air quality because of their theoretical foundation, diverse models, and accurate forecasting effects, such as multiple linear regression (Stadlober et al. 2008; Genc et al. 2010; Li et al. 2011), support vector machine (SVM) (Deng et al. 2018;

Responsible Editor: Constantini Samara

✉ Zhili Zhao
zhaozhl@lzu.edu.cn

[1] School of Information Science and Engineering, Lanzhou University, Lanzhou, 730000, China

Published online: 17 May 2020

Osowski and Garanty 2007), and artificial neural network (ANN) (Cabaneros et al. 2019; Perez and Reyes 2006; Feng et al. 2015). However, on the one hand, although such traditional models employed by some efforts are widely used and have reasonable performance in many domains, they are not suitable for handing time serial data since they cannot well process the time-steps of a sequence. In other words, they cannot well deal with the relationship between old information and new input in a sequence. On the other hand, such efforts do not yield the desired performance for air quality prediction. There are also efforts that employ deep learning models in air quality prediction, and the main solutions of adopting deep learning in air quality prediction are utilizing recurrent deep learning models, such as recurrent neural network (RNN) (Pineda 1987), long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997), and transferred bi-directional LSTM (Ma et al. 2019; Ma et al. 2019). Such models generate multi-layered representations of data, and also exhibit temporal dynamic behavior for time serial data, thus providing better performance over traditional machine learning methods. Moreover, there are also other efforts, such as Lin et al. who proposed a neuro-fuzzy network, in which the training data are described by fuzzy clusters with statistical means and variances to address the uncertainty of the involved impact factors (Lin et al. 2020). Jiang et al. presented a hybrid air quality prediction approach with pigeon-inspired optimization and extreme learning machine. The work employed a modified extreme learning machine to predict the data sub-series clustered based on the multidimensional scaling and K-means clustering (Jiang et al. 2019). Wang et al. proposed an ensemble deep learning model which considered both weather patterns and spatial-temporal properties (Wang and Song 2018). Maciaga et al. proposed a clustering-based ensemble model based on several evolving spiking neural networks on a separate set of time series for air quality prediction (Maciag et al. 2019). Compared with the efforts mentioned above, in this paper, we propose a hybrid ensemble model CERL to exploit the merits of both forward neural networks and recurrent neural networks for hourly air quality data prediction in Northwest of China. We take two cities in Northwest China, i.e., Lanzhou and Xi'an, as examples, and demonstrate the superiority of CERL. Moreover, we analyze the impact on the CERL performance as the step length that it can predict increases.

The rest of this paper is organized as follows. "Related work" presents a brief literature review on the work related to air quality prediction. In "Proposed approach," different prominent machine learning methods used for air quality prediction are presented. In addition, the hybrid method proposed in this paper is introduced. In "Materials," the materials used by this paper are given. In "Experiments and results," the results of the hourly air quality data are presented. "Discussions" discusses the CERL improvements in different step prediction and its superiority based on a hypothesis testing. "Conclusion" summarizes the achievements and highlights of this paper, and outlines directions for future work.

Related work

Air quality forecast predicts air pollution levels for a period ahead and provides important information to the public. However, the prediction is still a challenge because of the complexity of the process involved and the strong coupling across many parameters, which affect the modeling performance (Leksmono et al. 2006; Biancofiore et al. 2017). There have been three main types of air quality prediction methods: deterministic methods, statistical methods, and machine learning methods (Ma et al. 2019; Athira et al. 2018; Kwok et al. 2017; Singh et al. 2012). Deterministic methods usually build simulation models to simulate and predict the diffusion and transport process of atmospheric pollutants. But such methods often have large computation costs and low prediction accuracy if underlying atmospheric conditions are complex. Statistical methods are a kind of data-driven way of air quality prediction, and the most of statistical methods assumed the relationships between the input variables and the target outputs are linear (Ma et al. 2019), for example, multiple linear regression (Stadlober et al. 2008; Genc et al. 2010; Li et al. 2011). Such linear approaches suffer from the non-linearity of the real world. Machine learning-based methods often focus on nonlinear models, and the main methods fall into this category are ANN (Cabaneros et al. 2019; Perez and Reyes 2006; Feng et al. 2015), SVM (Deng et al. 2018; Osowski and Garanty 2007), etc. For example, Cabanerosa et al. reviewed the research activities in air pollution forecasting with ANNs and showed that feed-forward and hybrid ANN models with ad hoc optimization approaches were predominantly used to forecast long-term air pollutant factors (Cabaneros et al. 2019). Yang et al. presented a support vector regression model to predict $PM_{2.5}$ concentrations by considering spatial heterogeneity and dependence among the data (Deng et al. 2018). Note that there are also efforts that consider both statistical methods and machine learning methods as statistical methods (Mallet and Sportisse 2008; Zhang et al. 2012). Such linear and nonlinear data-driven methods usually build models fast with moderate accuracy, and have been studied a lot in recent years. For example, Singh et al. explored both linear and nonlinear approaches to predict air quality with the selected air pollutant factors and meteorological conditions as the estimators (Singh

et al. 2012). They argued that the nonlinear models, especially artificial neural network-based models and their variants, performed relatively better than linear PLSR models. Carcia predicted PM_{10} concentrations based on generalized linear models (GLMs), which focused on the relationship between atmospheric concentrations of air pollutants and meteorological variables (Garcia et al. 2016). In GLM, PM_{10} concentration was considered a dependent variable and both gaseous pollutants and meteorological variables were considered independent variables. Based on the similarity of $PM_{2.5}$ variation in monitoring network, He et al. proposed two methods, the linear method of stepwise regression and the nonlinear method of support vector regression, to predict $PM_{2.5}$ concentration (He et al. 2018). Shang et al. proposed a method on training local models based on a combination of classification and regression tree (CART) and ensemble extreme learning machine (EELM) to address the global-local duality and improve the prediction accuracy (Shang et al. 2019).

Besides the traditional methods based on machine learning algorithms, there are efforts that employ deep learning models in air quality prediction. Deep learning is a branch of machine learning that generates multi-layered representations of data, commonly using artificial neural networks, and has improved the state of the art in various machine learning tasks (Lang et al. 2019). The main solutions of adopting deep learning in air quality prediction are utilizing recurrent deep learning models, such as RNN (Pineda 1987) and LSTM (Hochreiter and Schmidhuber 1997). For example, Biancofiore et al. adopted a recurrent neural architecture, i.e., Elman Recurrent Network, to forecast daily averaged concentration of PM_{10}, and argued that RNN had better performances compared with both the multiple linear regression model and the neural network model without the recursive architecture (Biancofiore et al. 2017). In Athira et al. (2018), Athira V compared different RNN models and their variations based on the pollution and meteorological time series AirNet data (Zhao et al. 2018), and showed that the performance of gated recurrent unit network was slightly higher than that of RNN and LSTM networks. Ma et al. used a bi-directional LSTM model to learn long-term dependencies of $PM_{2.5}$ (Ma et al. 2019). The highlight of the work was the combination of a bi-directional LSTM and transfer learning technique, which could transfer the knowledge from smaller temporal resolutions to larger ones. Based on the work, Ma et al. also proposed a stacked bi-directional LSTM that combined deep learning techniques and transfer learning to deal with the data shortage problem (Ma et al. 2019).

In addition, there are hybrid models which exploit the advantages of multiple models, such as a hybrid model based on sample entropy, secondary decomposition, and least squares support vector machine LSTM AQI prediction (Wu and Lin 2019). Wang et al. proposed a deep spatial-temporal ensemble model, which considered not only meteorological information but also spatial and temporal properties to predict air quality. LSTM was also used to learn both short-term and long-term dependencies (Wang and Song 2018).

To sum up, there are a variety of differences between the aforementioned efforts and our work. Our work is a kind of ensemble model to exploit the merits of both forward neural networks and recurrent neural networks that are designed for handling time serial data. Based on the advantages of both different types of neural networks, CERL provides better performance over baseline models. In particular, we focus on the air quality prediction of two rarely studied capital cities in Northwest of China, and build prediction models for main pollutant factors, i.e., AQI (AirNow 2019), $PM_{2.5}$, PM_{10}, CO, SO_2, NO_2, and O_3 hours by hours.

Proposed approach

In this work, we combined forward neural networks with several recurrent neural networks as a hybrid model with an aim to improve the accuracy of air quality prediction. This section first introduces several machine learning methods that are often used for air quality prediction, and then introduces our hybrid combined approach CERL.

Prominent approaches for time series data

Cascade-forward neural network

Cascade-forward neural network (CFNN) is an artificial neural network in which the information moves only forward, i.e., from the input nodes, through the hidden nodes to the output nodes. Moreover, CFNN includes a connection from the input and every previous layer to following layers. In other words, in a CFNN with three layers, the output layer is also connected directly with the input layer except hidden layer, as shown in Fig. 1a. As with feed-forward networks, CFNNs with single hidden layer can arbitrarily closely approximate any continuous function that maps intervals of real numbers to some output interval of real numbers. Based on the direct connection between the input and output, CFNN is often used for time series prediction. For example, Tengeleng et al. utilized a cascade forward back-propagation neural network (BPNN) to predict rain parameters, i.e., water content, rain rate, and radar reflectivity with raindrop size distribution (Tengeleng and Armand 2014). Warsito et al. showed that CFNN models could successfully predict both simulated time series data and monthly palm oil price index data (Warsito et al. 2018, 05).

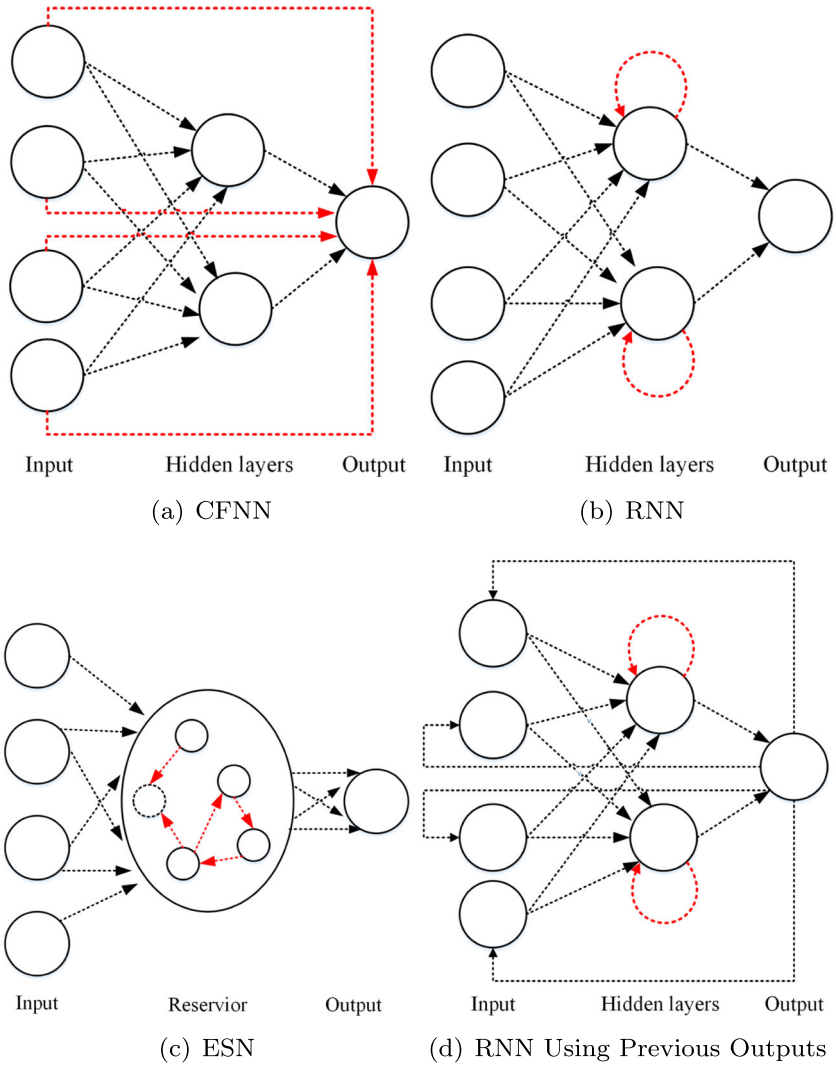


Fig. 1 The models employed for air quality prediction

RNN

RNN is a kind of artificial neural network that is specially designed to model time serial data. Unlike feed-forward networks, the hidden layers of RNN are connected back into themselves to maintain an internal state and allow RNN to exhibit temporal dynamic behavior for a time sequence, as shown in Fig. 1b. Therefore, RNN enables the networks to do temporal processing, and Biancofiore et al. argued that RNN had better performances compared with other neural network models without the recursive architecture on forecasting daily averaged concentration of PM_{10} (Biancofiore et al. 2017).

ESN

Roughly speaking, echo state network (ESN) is a special case of recurrent neural network with a non-trainable sparse random recurrent part (reservoir) and a simple linear readout (Jaeger 2001, 01), as shown in Fig. 1c. Connection weights in the ESN reservoir, as well as the input weights, are randomly generated. Compared with other RNN models, ESNs can efficiently process the temporal dependency of time series with high nonlinear mapping capacity and dynamic memory (Shen et al. 2016; Lukoševičius and Jaeger 2009).

Recurrent networks using previous outputs

Besides the standard recurrent neural networks, in which each layer has a recurrent connection with a tap delay associated with it, there are variant RNNs that have delayed recurrent connections between their output and the input layer, as shown in Fig. 1d. In such networks, the state of the model is influenced not only by its previous internal states but also by its outputs. This is useful in modeling time serial data, since the output for timestep t is helpful to predict the output for timestep $t + d$, where d is the step length of the time serial prediction.

Proposed hybrid approach

As we can see from above sections, recurrent neural networks are a powerful type of artificial neural networks, in which the outputs of hidden layers are fed back into the same hidden layer. Such kind of internal memory is helpful for handling time serial data, i.e., the data that occurs in a time sequence. In this paper, we focus on combining forward neural networks with prominent recurrent neural networks as a hybrid model CERL with an aim to improve the accuracy of air quality prediction. The general process of building the hybrid model has two stages: single model learning and hybrid model learning, as shown in Fig. 2.

As other supervised learning algorithms, we split the data set into two sets: training and test sets, which are used to fit a model and assess the model at the end of the model building, respectively. In the first stage, several single recurrent neural network models are built based on mapping input features to output labels. Such models need to be optimized to have their best performance. After the optimized single models are built, they are used to calculate the predictions to the training set. The predictions to the training set are denoted by $train\_Y_1'$, $train\_Y_2'$, and $train\_Y_n'$. Accordingly, the predictions to the test are denoted by $test\_Y_1'$, $test\_Y_2'$, and $test\_Y_n'$. In the general model building process of a supervised learning algorithm, such prediction results are used to calculate training error and test error. In our work, such prediction results are grouped together as the features of training and test sets to build the hybrid ensemble model, respectively. It is worth noticing that the labels of the training and test sets used to building single models are reused. In other words, the goal of the hybrid ensemble model is to map the intermediate prediction results of the single models to the final output labels of the training set. Such a regression process can be implemented by many machine learning algorithms, such as linear regression, BPNN, and SVM.

Since artificial neural network has been well established by many successful applications in a variety of fields (Yoon et al. 2011; Singh et al. 2012), in this work, we employed a three-layer BPNN for our hybrid ensemble model. We used the logistic sigmoid function as the activation functions on hidden neurons, which is defined as follows,

$$f(y_j) = \frac{1}{1 + e^{-y_j}} \tag{1}$$

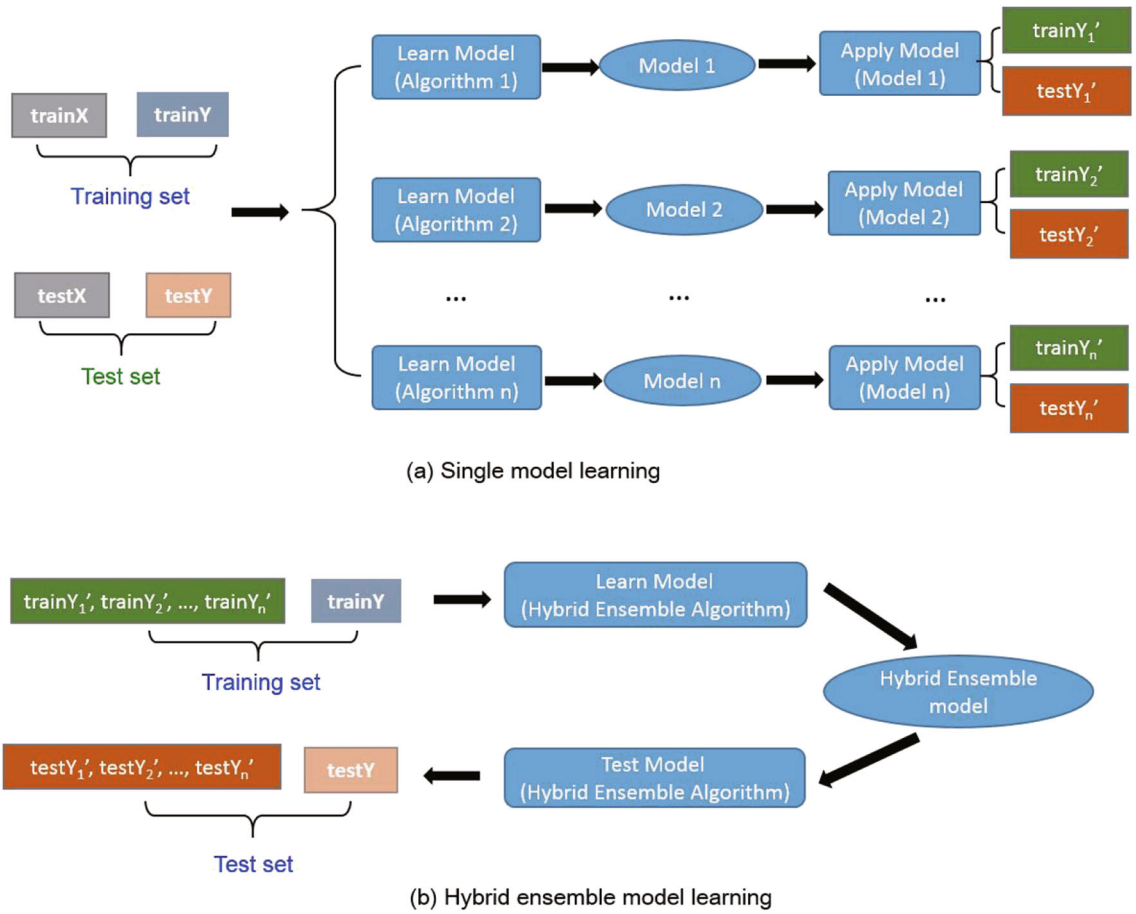


Fig. 2 The general process of building the hybrid ensemble model

where y_j is the output of a hidden neuron i, which is calculated as follows,

$$y_j = \sum_{i=1}^{n} w_{ij} train\_Y_i' + \theta_j \quad (2)$$

where $train\_Y_i'$ is the output of the single models, and it is used as the input of the ensemble model. w_{ij} is the weight from input neuron i to neuron j, and θ_j is the bias of neuron i.

At the output layer, we used mean square error (MSE) as the loss function, which is defined as follows,

$$\min_{w} J(w) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{y}_i)^2 \quad (3)$$

where Y_i is the actual output of training instance i and $\hat{y}_i$ is the output from the neural network for the instance i. Our goal is to minimize the loss function J as the neural network is trained.

Materials

Datasets

We evaluated the performance of our proposed model on a set of air quality data which are extracted from the web site of historical data of air quality in China (Wang 2019), which provides download services of historical air quality data for all cites in China since May 13, 2014. The air quality data was from China Environmental Monitoring Station (CNEMC 2019), which updates the data daily. The air pollutant factors include AQI, $PM_{2.5}$, PM_{10}, CO, SO_2, NO_2, and O_3 hours by hours. Moreover, the data also includes the average values of $PM_{2.5}$, PM_{10}, CO, SO_2, NO_2, and O_3 over a 24-h period.

We selected the air quality data of two capital cities in Northwest of China, i.e., Xi'an and Lanzhou. We selected the dataset which was from January 1–31, 2019, since both cities often have the worst air quality in December and January, as shown in Fig. 3, which shows the monthly average AQI and $PM_{2.5}$ values of Xi'an and Lanzhou over 69 months from January 2014 to August 2018. Note that the data of Fig. 3 was from China Air Quality Online Monitoring and Analysis Platform (Wang 2019). The data is collected once in an hour; finally, we got 744 (24 × 31) samples for each given city in the first month of 2019. Besides the date and time, each sample also includes 6 factors given a specified hour, i.e., AQI, the concentration of $PM_{2.5}$, PM_{10}, CO, SO_2, NO_2, and O_3.

Data preprocessing

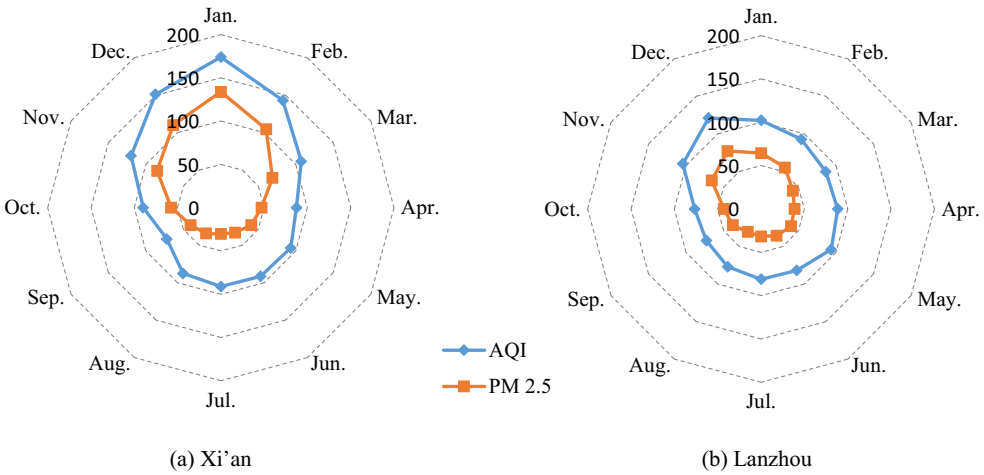
Before building machine learning models for air quality prediction, we processed data by filling missing values, handing noisy data, normalization, and dataset split. The details of such processes will be presented in this section.

Filling missing values

Due to technical and other reasons, a small amount of the air quality data provided by the Environment Cloud were missing. This is very common in environment monitoring, but may have a significant effect on the conclusions that are drawn from the data.

Since our data were in time serials, we filled missing values based on linear extrapolation, which is a method used to construct new data points based on a discrete set of known data points. For example, if the data of two samples are given by two coordinates (t_1, y_1) and (t_2, y_2), the missing data y_* at the time t_* are calculated with the following formula:

$$y_* = y_1 + \frac{(t_* - t_1)}{(t_2 - t_1)}(y_2 - y_1). \quad (4)$$



(a) Xi'an (b) Lanzhou

Fig. 3 The air quality of Xi'an and Lanzhou (Jan. 2014–Aug. 2018)

Table 1 The data statistical details

City	Indicator	Air pollutant factors						
		AQI	$PM_{2.5}$ ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	SO_2 ($\mu g/m^3$)	NO_2 ($\mu g/m^3$)	O_3 ($\mu g/m^3$)	CO (mg/m^3)
Lanzhou	Max.	160.00	97.00	87.00	175.00	150.00	41.00	35.00
	Min.	38.00	16.00	20.00	37.00	46.00	5.00	10.00
	Avg.	73.35	45.89	47.15	87.13	86.37	24.28	24.96
Xi'an	Max.	308.00	233.00	227.00	375.00	340.00	24.00	22.00
	Min.	53.00	9.00	20.00	54.00	64.00	4.00	5.00
	Avg.	172.97	127.96	133.03	182.95	183.72	15.85	16.33

In other words, the data y_1, y_2, y_* are in a straight line. Note that t_* can be within or outside the time interval $[t_1, t_2]$.

In this step, we filled 7 missing values, and the statistical details of each pollutant factor after filling missing values can be found in Table 1.

Noise reduction based on singular spectrum analysis

In machine learning, noisy data caused by different erratic factors usually affect the forecast accuracy. To deal with such problem, we employed singular spectrum analysis (SSA) to handle the noisy data. SSA is a model-free method and can be used to decompose original series into a sum of interpretable components, such as trend, periodic components, and noise. Afterwards, the signals can be extracted from noisy data by discarding some decomposed components. In other words, the noise reduction data is obtained by adding the first several decomposed components together.

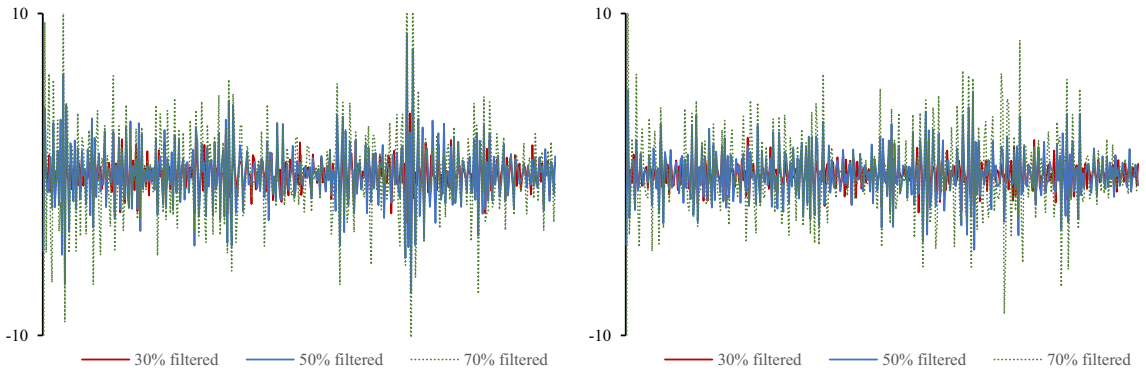
In the practical application of SSA, the optimal number of the data reconstruction is usually the half length of decomposed components (He et al. 2019). In our work, the data series were decomposed into 100 components, and different numbers of the components ranging from 10 to 70 were regarded as noises and discarded to evaluate the denoisy performance. Generally speaking, the components discarded result in a smoother and slower varying data series. Figure 4 presents the noise reduction residuals of AQI and $PM_{2.5}$ of Lanzhou from January 1 to January 15 by different percentages. The residuals are the differences between the original values and the values after the noise reduction. We can see that the residuals become larger as more components are regarded as noises and discarded. However, the noise reduction does not change the trend of the data since the residuals are often very small, and the only difference is that the curve becomes smoother as more data are reduced. The performance evaluation of data denoise will be detailed in "Denoise evaluation."

Normalization

The dataset in our work was normalized by the *mapminmax* function of MATLAB, which is defined as:

$$mapminmax(X, y_{min}, y_{max}) = \frac{(y_{max} - y_{min}) * (x - x_{min})}{x_{max} - x_{min}} + y_{min} \quad (5)$$

where X is the matrix to be normalized; y_{min} and y_{max} are expected minimum and maximum values of each row of X, respectively; and x_{min} and x_{max} are actual minimum and



(a) AQI (b) $PM_{2.5}$

Fig. 4 The noise reduction residuals by SSA

maximum values of each row of X. In our work, the dataset is normalized to [0.001,1].

Dataset split

In our work, we took 7 pm on January 24, 2019, as the time point to split training and test sets, and ensured that they contained 80% and 20% instances, respectively. Moreover, we further split the time serial training and test datasets based on a sliding window algorithm, which was used to segment a collection of historical air quality data into groups. The algorithm procedure can be found in Algorithm 1.

Algorithm 1 Splitting time serial data.

Require: : data, $window\_size$, $step\_length$
Ensure: : X, Y
function $sliding\_window(data,\ window\_size,\ step\_length)$
  $d \leftarrow length(data)$;
  $window\_num \leftarrow d - window\_size - step\_length + 1$;
  for i from 1 to $window\_num$ **do**
    $window \leftarrow data(i : i + window\_size - 1)$;
    $X \leftarrow [X; window]$;
    $step \leftarrow data(i + window\_size : i + window\_size + step\_length - 1)$;
    $Y \leftarrow [Y; step]$;
  end for
  return X, Y;
end function

where *data* is one-by-d matrix, d is the length of the data, *window_size* is the number of consecutive observations per sliding window, and *step_length* is the number of steps ahead to forecast. The algorithm takes the data, window size and step length as the input, and outputs X and Y, which are then used to learn target air quality prediction models. For example, suppose a time sequence is denoted as data = $(s_1, s_2, \ldots, s_{100})$, *window_size* = 3 and *step_length* = 1, then $X = (\langle s_1, s_2, s_3\rangle, \langle s_2, s_3, s_4\rangle, \ldots, \langle s_{97}, s_{98}, s_{99}\rangle)$ and $Y = (s_4, s_5, \ldots, s_{100})$. If *window_size* = 5 and *step_length* = 2, then $X = (\langle s_1, s_2, s_3, s_4, s_5\rangle, \langle s_2, s_3, s_4, s_5, s_6\rangle, \ldots, \langle s_{94}, s_{95}, s_{96}, s_{97}, s_{98}\rangle)$ and $Y = (\langle s_6, s_7\rangle, \langle s_7, s_8\rangle, \ldots, \langle s_{99}, s_{100}\rangle)$. Note that the slide step of our sliding window algorithm is 1.

Evaluation methodology metrics

In this paper, the following three metrics were employed to evaluate the performance of the involved models. There are mean absolute deviation (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), and correlation coefficients (R), which are calculated with the following formulas.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{6}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - \hat{x}_i\right| \tag{7}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right| \times 100\% \tag{8}$$

$$R = \frac{n\sum_{i=1}^{n}(x_i\hat{x}_i) - (\sum_{i=1}^{n}x_i)(\sum_{i=1}^{n}\hat{x}_i)}{\sqrt{n\sum_{i=1}^{n}x_i{}^2 - (\sum_{i=1}^{n}x_i)^2}\sqrt{n\sum_{i=1}^{n}\hat{x}_i^2 - (\sum_{i=1}^{n}\hat{x}_i)^2}} \tag{9}$$

where x_i and $\hat{x}_i$ represent the actual value and the predicted value, respectively, and n is the number of test samples.

The model training parameters

In this paper, our hybrid ensemble model CERL combined forward neural networks with recurrent neural networks that are designed for handling time serial data to predict air quality in Lanzhou and Xi'an. More precisely, we took CFNN, RNN, ESN, and RNN using previous outputs as baseline models, and combined them using BPNN to improve the prediction performance. To demonstrate the superiority of CERL over such baseline models, both baseline models and CERL were optimized to compare their best performance. For CFNN and RNN, we used MATLAB functions *cascadeforwardnet* and *layrecnet* for CFNN and RNN implementation, respectively. We specified the number of hidden neurons of CFNN and RNN as $\log_2 n$, where n is the size of the input layer. We used the ESN MATLAB library developed by Jaeger et al. (2007). The number of internal units was set to $n * n$, where n is the size of the input layer. The spectral radius of the ESN reservoir was 0.01 to ensure that the ESN had the echo state property. We used Pyrenn (Atabay 2019), which is a recurrent neural network toolbox for python and MATLAB for the implementation of RNN using previous outputs. As with CFNN and RNN, we specified the number of hidden neurons of Pyrenn as $\log_2 n$, and the number of output delays as 2. Note that we used Pyrenn as the name for the RNN models using previous outputs for short in what follows. Moreover, in CERL, we used BPNN to combine the prediction results of CFNN, RNN, ESN, and Pyrenn. We specified the BPNN learning parameters as follows:

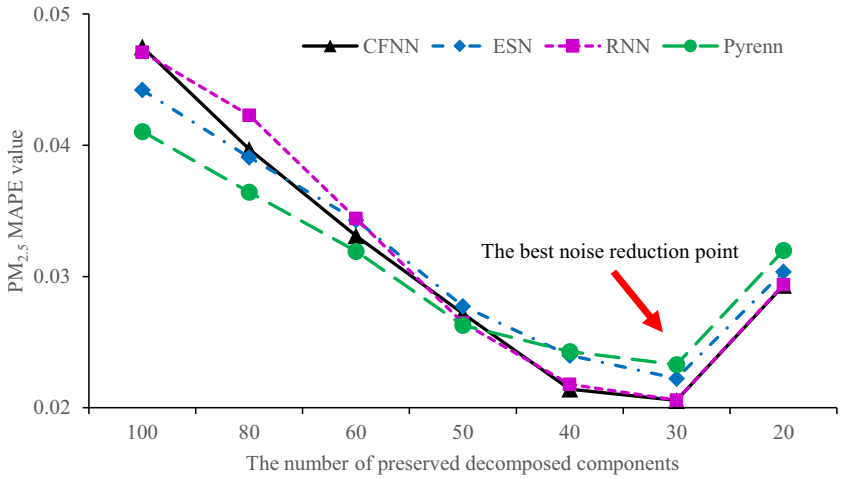


Fig. 5 Denoised by different percentages

learning rate 0.001, maximum number of iterations 2000, and the number of hidden neurons 4. Moreover, we used "trainbr" as the training function to avoid overfitting, since it usually worked well with early stopping. In order to make the comparison more reasonable, the best performance of all aforementioned models was the average of 100 times of training and testing such models.

Experiments and results

Denoise evaluation

As mentioned in "Noise reduction based on singular spectrum analysis," the noise reduction does not change the trend of the data, but makes the curve smoother as

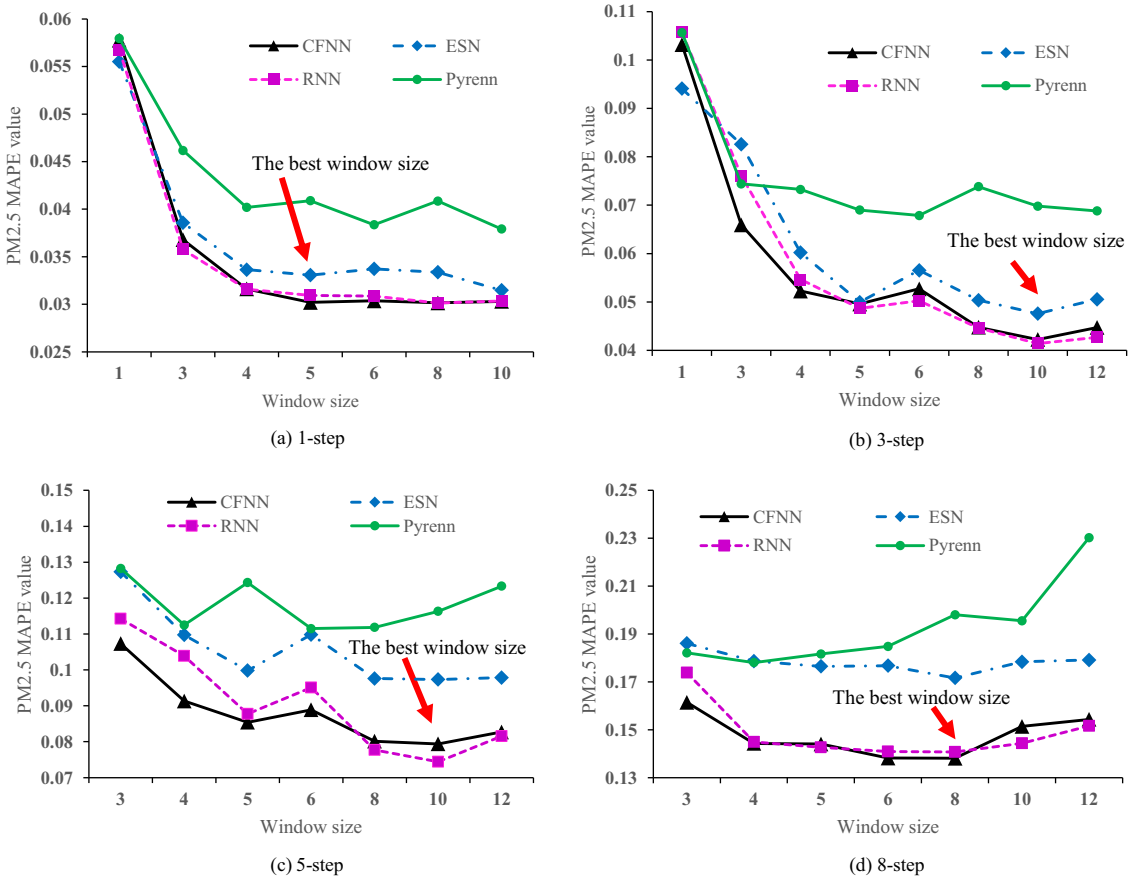


Fig. 6 *N*-step window size decision

more data are reduced. Moreover, it is helpful to improve the model performance after the noisy data are removed. In order to demonstrate the denoise performance, we used the dataset that was reduced by different percentages as the input of different baseline models, i.e., CFNN, RNN, ESN, and Pyrenn to build 1-step $PM_{2.5}$ prediction models for Lanzhou. Figure 5 shows the performances of different models. In the SSA of our work, the data series were decomposed into 100 components. As we can see, the performance of different models improves as the more data are reduced, and the best performance is obtained when the noise reduction percentage is 70%. In other words, the noise reduction data is obtained by adding the first 30 components together. The performance of all models goes along the same trend, which proves that the denoise reduction is useful to improve the model performance. In the following experiments, the noise reduction data is obtained by adding the first 30 components together except clearly specified.

1-step prediction

1-step prediction means that the models can be used to predict air quality for the next hour. To get an optimal model, we used different window sizes to split the training set and the test set. The optimal window size was obtained by analyzing the prediction performance of $PM_{2.5}$, and then, it was used to build the models for other pollutant factors, i.e., AQI, PM_{10}, CO, SO_2, NO_2, and O_3.

Window size decision

In this paper, we used a sliding window algorithm to split training and testing sets. In order to find an appropriate window size, we used different window sizes ranging from 1 to 10 to prepare the training set and the test set. Different models, i.e., CFNN, RNN, ESN, and Pyrenn, used such data as the input to build 1-step $PM_{2.5}$ prediction models

Table 2 The performance of different models for 1-step prediction in Lanzhou

Indicator	Metrics	CFNN	ESN	RNN	Pyrenn	CERL	Average value
AQI	MAPE	0.0204	0.0222	0.0205	0.0237	*0.0201*	0.0214
	MAE	1.2265	1.3415	1.2282	1.4251	*1.2115*	1.2866
	RMSE	1.6044	2.2555	1.6047	1.9237	*1.5848*	1.7946
	R	0.9836	0.9674	0.9836	0.9774	*0.9838*	0.9792
$PM_{2.5}$	MAPE	0.0300	0.0329	0.0310	0.0376	*0.0294*	0.0322
	MAE	0.8939	1.0051	0.9176	1.1287	*0.8785*	0.9648
	RMSE	1.2217	1.8687	1.2435	1.5469	*1.2115*	1.4185
	R	0.9842	0.9640	*0.9839*	0.9750	0.9844	0.9783
PM_{10}	MAPE	0.0343	0.0361	0.0343	0.0395	*0.0337*	0.0356
	MAE	2.3192	2.4577	2.3200	2.6472	*2.2973*	2.4083
	RMSE	3.0442	3.5454	*3.0368*	3.4939	3.0395	3.2320
	R	0.9832	0.9772	*0.9833*	0.9792	0.9833	0.9812
SO_2	MAPE	0.0890	0.0900	0.0893	0.0959	*0.0872*	0.0903
	MAE	1.5267	1.5491	1.5310	1.6160	*1.5223*	1.5490
	RMSE	*2.0323*	2.0652	2.0356	2.0931	2.0453	2.0543
	R	*0.9817*	0.9811	0.9817	0.9811	0.9816	0.9814
NO_2	MAPE	*0.0488*	0.0490	0.0493	0.0595	0.0532	0.0520
	MAE	*1.6621*	1.6781	1.6696	1.9906	1.7379	1.7477
	RMSE	*2.2425*	2.2736	2.2487	2.5726	2.3326	2.3340
	R	*0.9873*	0.9869	0.9872	0.9842	0.9866	0.9864
O_3	MAPE	0.0484	0.0492	0.0483	0.0593	*0.0463*	0.0503
	MAE	1.8998	1.9370	1.8988	2.2627	*1.8576*	1.9712
	RMSE	2.5321	2.5709	2.5304	3.3173	*2.4807*	2.6863
	R	0.9900	0.9897	0.9900	0.9828	*0.9905*	0.9886
CO	MAPE	0.0628	0.0645	0.0628	0.0731	*0.0620*	0.0650
	MAE	0.0533	0.0547	0.0531	0.0626	*0.0527*	0.0553
	RMSE	0.0748	0.0777	*0.0746*	0.0887	0.0746	0.0781
	R	0.9781	0.9763	*0.9783*	0.9690	0.9782	0.9760

for Lanzhou. Figure 6 a shows the performance of different models. We can see that the performance improves as the window size increases, and the curves become flat after the window size is bigger than 5. Therefore, in our 1-step air quality prediction models, we took 5 as the window size. It is worth noticing that some models had slight better performance when the window size was bigger than 5, but we still specified the window size as 5 to reduce computing costs. As a result, the dataset was divided into a training set of 590 samples and 144 test samples.

Performance comparison

To illustrate the performance of CERL, we used CFNN, RNN, ESN, and Pyrenn as baselines to build 1-step models for 7 air pollutant factors including AQI, $PM_{2.5}$, PM_{10}, CO, SO_2, NO_2, and O_3. Afterwards, we compared the performances of such models with CERL. The noise reduction data in such experiments was obtained by adding the first 30 components together, and the window size was 5. Each model was optimized to get its best performance. Moreover, each model was trained and tested 100 times to get its average performance. Tables 2 and 3 show the performance results of different models in Lanzhou and Xi'an, respectively. Note that the best results are indicated in italics.

We can see that all such models have good performance to predict air quality in both Lanzhou and Xi'an. The average MAPE values of such models for AQI, $PM_{2.5}$, PM_{10}, SO_2, NO_2, O_3, and CO in Lanzhou are 2.14%, 3.22%, 3.56%, 9.03%, 5.20%, 5.03%, and 6.50%, respectively. In Xi'an, the average MAPE values of such models for AQI, $PM_{2.5}$, PM_{10}, SO_2, NO_2, O_3, and CO are 2.56%, 2.70%, 2.87%, 4.98%, 2.98%, 8.96%, and 2.38%,

Table 3 The performance of different models for 1-step prediction in Xi'an

Indicator	Metrics	CFNN	ESN	RNN	Pyrenn	CERL	Average value
AQI	MAPE	0.0231	0.0235	0.0232	0.0360	*0.0225*	0.0257
	MAE	2.9440	3.0209	2.9496	4.3282	*2.9138*	3.2313
	RMSE	*4.1602*	4.4751	4.1676	5.5782	4.1708	4.5104
	R	0.9956	0.9946	0.9956	0.9926	*0.9956*	0.9948
$PM_{2.5}$	MAPE	0.0228	0.0234	0.0228	0.0442	*0.0220*	0.0270
	MAE	2.0976	2.2019	2.1022	3.6397	*2.0668*	2.4216
	RMSE	3.0998	3.5338	3.1118	4.6620	*3.0660*	3.4947
	R	0.9967	0.9957	0.9967	0.9940	0.9968	0.9960
PM_{10}	MAPE	0.0259	0.0265	0.0259	0.0402	*0.0251*	0.0287
	MAE	3.1840	3.2931	3.1771	4.6986	*3.1391*	3.4984
	RMSE	*4.0692*	4.4762	4.0710	5.7516	4.0967	4.4929
	R	*0.9958*	0.9949	0.9958	0.9918	0.9957	0.9948
SO_2	MAPE	0.0452	*0.0447*	0.0489	0.0600	0.0504	0.0498
	MAE	*0.5805*	0.6187	0.5948	0.7262	0.6010	0.6242
	RMSE	*0.7790*	0.9648	0.7875	0.9799	0.7988	0.8620
	R	0.9933	0.9892	*0.9933*	0.9905	0.9933	0.9919
NO_2	MAPE	*0.0275*	0.0296	0.0277	0.0363	0.0279	0.0298
	MAE	*1.2942*	1.4469	1.2972	1.6259	1.3020	1.3932
	RMSE	1.6291	2.6565	*1.6289*	2.1214	1.6341	1.9340
	R	0.9963	0.9899	*0.9964*	0.9944	0.9963	0.9947
O_3	MAPE	0.0837	0.0985	0.0843	0.0996	*0.0819*	0.0896
	MAE	*1.2627*	1.4966	1.3093	1.5361	1.3278	1.3865
	RMSE	*1.6767*	2.7924	1.8892	2.009	1.7625	2.0260
	R	*0.9965*	0.9903	0.9955	0.9951	0.9963	0.9947
CO	MAPE	0.0209	0.0213	0.0210	0.0352	*0.0206*	0.0238
	MAE	0.0259	0.0267	0.0260	0.0410	*0.0256*	0.0290
	RMSE	*0.0337*	0.0377	0.0339	0.0536	0.0338	0.0385
	R	*0.9946*	0.9931	0.9945	0.9892	0.9946	0.9932

respectively. Moreover, among such models, CERL exhibits an improvement over CFNN, ESN, RNN, and Pyrenn on 6 of 7 air pollutant factors in both Lanzhou and Xi'an. In Lanzhou, CERL provides superior performance on all pollutant factors except NO_2 prediction. For example, the MAPE value of CERL for the AQI prediction in Lanzhou is 2.01%, CFNN has the second smallest MAPE value 2.04%, and Pyrenn has the worst MAPE value 2.37%. But the MAPE value of CERL for the NO_2 prediction is 5.32%, which is bigger than that of CFNN with the best MAPE value of 4.88%. In Xi'an, CERL does not get the best performance only on SO_2 prediction. For example, the MAPE value of CERL for the AQI prediction in Xi'an is 2.25%, CFNN has the second smallest MAPE value 2.31%, and Pyrenn has the worst MAPE value 3.60%. But the MAPE value of CERL for the SO_2 prediction is 5.04%, which is bigger than that of ESN with the best MAPE value 4.47%. However, although CERL has superior performance over other models, the performance of such models is similar. This is because such models are adequate for predicting air quality precisely in a short term. Figure 7 shows the comparison of such models on MAPE metric.

The final prediction results of CERL in Lanzhou is given in Fig. 8. We can see that all such models have adequate performance and the forecasting values and the actual values are fitting very well on all pollutant factors.

N-step prediction

To demonstrate the performance of CERL on long-term air quality prediction, this section provides the comparison between CERL and the baseline models for air quality prediction in the next 3, 5, and 8 h, respectively. Note that the values in *N* steps ahead are simultaneously predicted by the models in our work, rather than based on the results of 1-step prediction. The datasets used to prepare the training and test sets can be found in Algorithm 1.

3-step prediction

As the 1-step prediction, we first did experiments to decide the window size of 3-step prediction, and Fig. 6 b shows the performance of different models on different window sizes to predict $PM_{2.5}$ in Lanzhou. We can see that the performance of all models except Pyrenn ups to optima as the window size is 10. As a result, the dataset is divided into 583 samples for the training set and 137 samples for the testing set. Pyrenn does not have the best performance but an approximate optimal value with the window size 10. Therefore, we used 10 as the window size to build 3-step air quality prediction models, and the results are shown in Fig. 9.

We can see that, unlike the 1-step prediction, CERL provides better performance on all air pollutant factors in both Lanzhou and Xi'an. Moreover, CERL has more obvious improvement than the other three baseline models. For example, the MAPE value of CERL for the 1-step $PM_{2.5}$ prediction is 2.94%, which only improves 2.04% over CFNN that has the second best performance with the MAPE value 3.00%. However, in the 3-step AQI prediction in Lanzhou, the MAPE value of CERL for the $PM_{2.5}$ prediction is 3.92%, which improves 7.98% over RNN with the second best MAPE value 4.26%. It is also true for the SO_2 prediction. In Xi'an, the MAPE value of CERL for the 1-step SO_2 prediction is 5.04%, which is even worse than that of CFNN, ESN, and RNN. However, in the 3-step AQI prediction in Xi'an, the MAPE value of CERL for the SO_2 prediction is 6.89%, which improves 11.67% over ESN with the second best MAPE value 7.80%.

5-step prediction

In the 5-step prediction, the best prediction performance is obtained when the window size is 10, as shown in Fig. 6c. We can see that the ESN and Pyrenn models do not have the

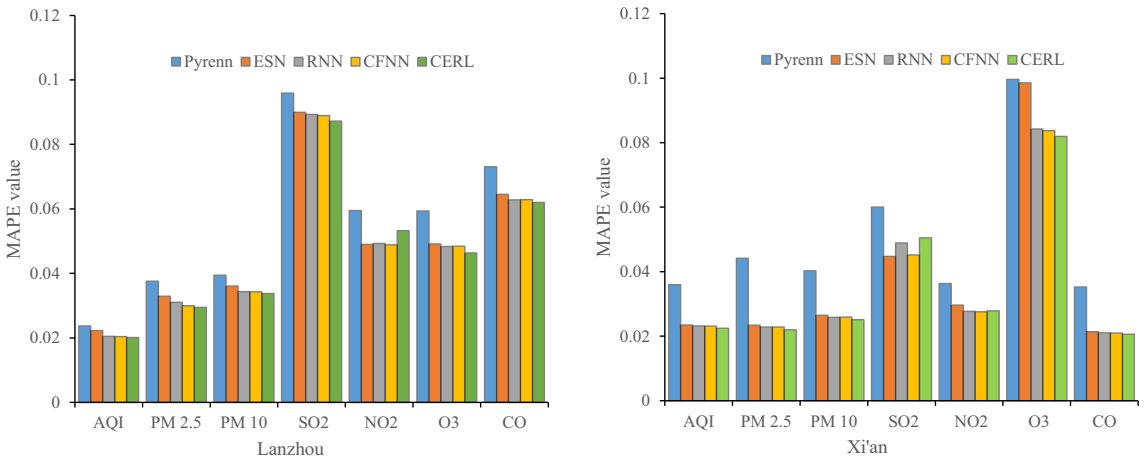


Fig. 7 1-step comparison (MAPE)

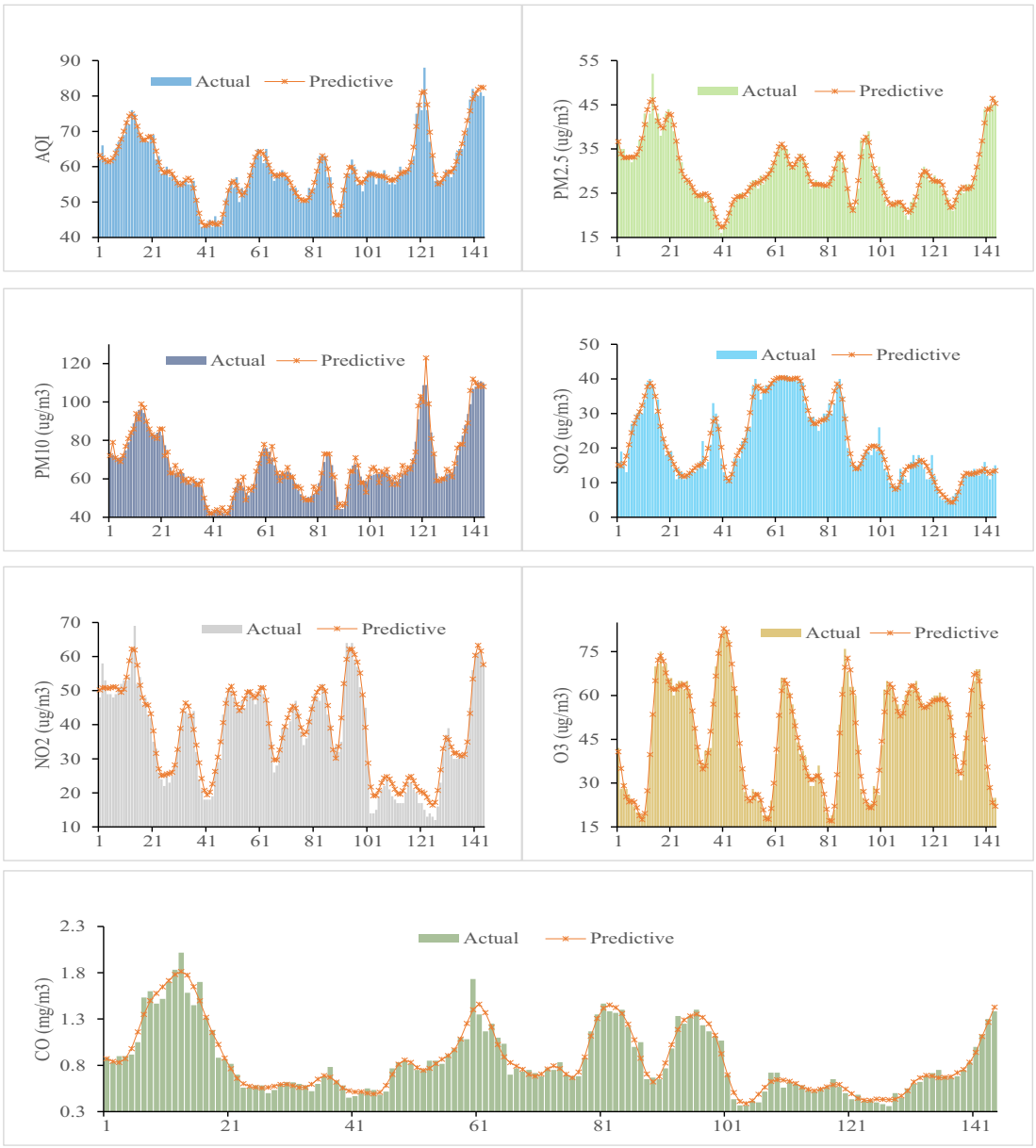


Fig. 8 CERL 1-step prediction in Lanzhou

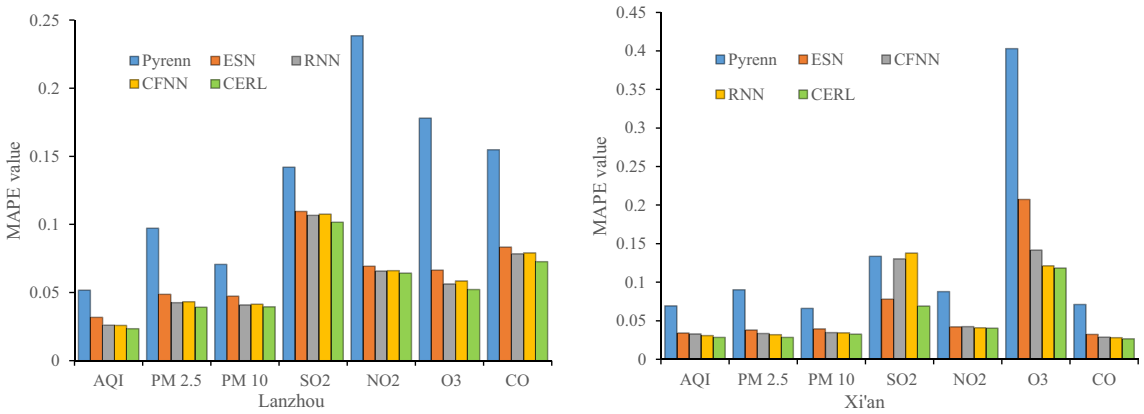


Fig. 9 3-step comparison (MAPE)

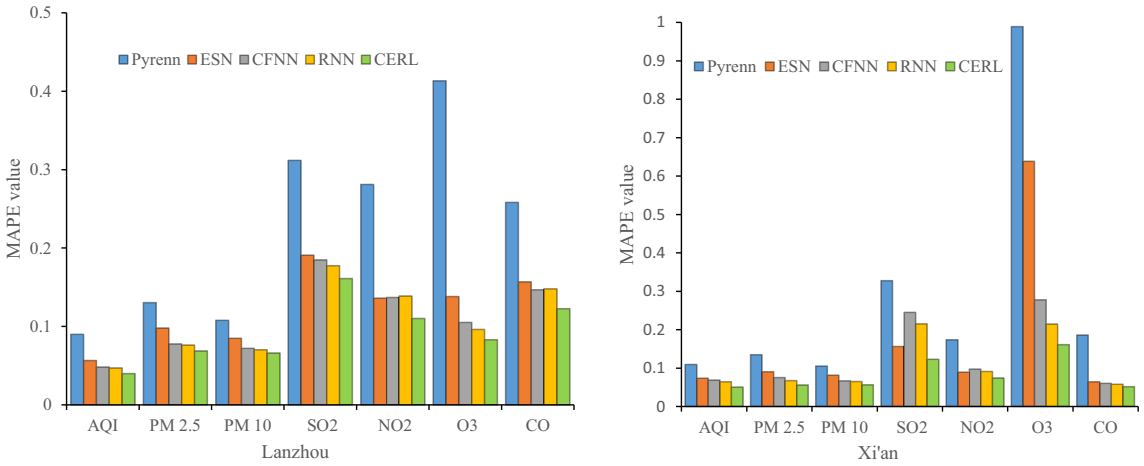


Fig. 10 5-step comparison (MAPE)

best performance, but they have approximate optimal value with the window size 10. As a result, the dataset is divided into 581 samples for the training set and 135 samples for the testing set. The performance of different 5-step models with the window size 10 is presented in Fig. 10.

As in the 3-step prediction, CERL has obvious better performance than in the other baseline models in all air pollutant factors, and the improvement is more clear than both 1-step and 3-step prediction. For example, the MAPE value of CERL for the 3-step $PM_{2.5}$ prediction is 3.92%, which improves 7.98% over RNN with the second best MAPE value 4.26%. However, in the 5-step $PM_{2.5}$ prediction in Lanzhou, the MAPE value of CERL for the $PM_{2.5}$ prediction is 6.87%, which improves 9.96% over RNN with the second best MAPE value 7.63%. In Xi'an, the MAPE value of CERL for the 3-step SO_2 prediction is 6.89%, which improves 11.67% over ESN with the second best MAPE value 7.80%. However, in the 5-step SO_2 prediction in Xi'an, the MAPE value of CERL for the SO_2 prediction is 12.30%, which improves 21.36% over ESN with the second best MAPE value 15.64%. However, as the step length increases, the overall performance declines. As shown in Figs. 7 and 9, the MAPE values of CERL 1-step and 3-step prediction fall in the range of 2.01∼8.72% and 2.66∼11.84% for almost all air pollutant factors, respectively. As the step length is increased to 5, the MAPE values of CERL fall in the range of 5.05∼16.12%.

8-step prediction

In our 8-step air quality prediction models, we took 8 as the window size, as shown in Fig. 6d. We can see that the best performance is achieved when the window size is 6, and it even declines when the window size is bigger than 10. As a result, the dataset is divided into 585 samples for the training set and 139 samples for the testing set.

In the 8-step prediction, CERL also has the best performance in almost all air pollutant factors prediction, as shown in Fig. 11. The MAPE value of CERL for the 8-step $PM_{2.5}$ prediction in Lanzhou is 10.97%, which

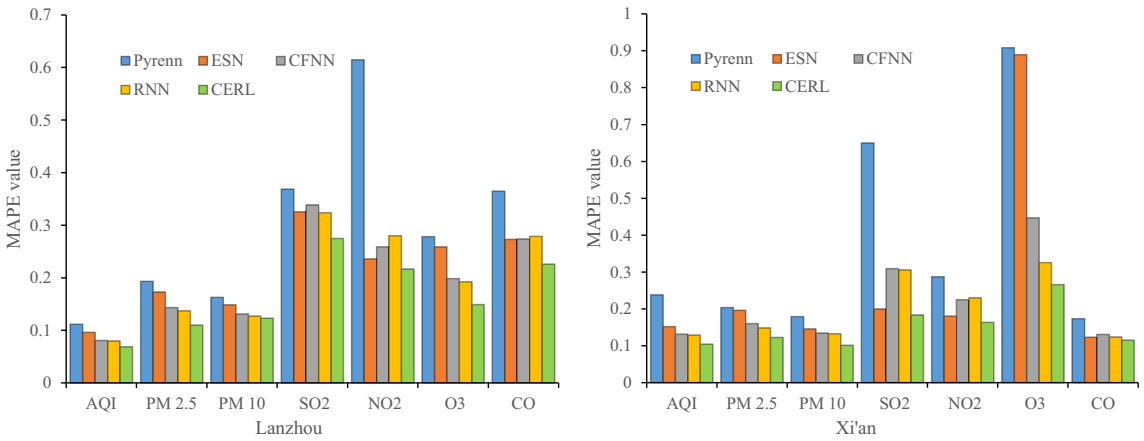


Fig. 11 8-step comparison (MAPE)

Table 4 The CERL MAPE improvements in different step prediction in Lanzhou

	AQI	$PM_{2.5}$	PM_{10}	SO_2	NO_2	O_3	CO
1-step	1.24%	1.82%	1.52%	1.98%	− 9.00%	4.12%	1.24%
3-step	9.00%	8.01%	3.17%	4.68%	2.54%	7.18%	7.47%
5-step	15.45%	9.98%	5.61%	9.17%	19.04%	13.47%	16.54%
8-step	13.88%	20.03%	3.16%	15.09%	8.15%	22.48%	17.19%

improves 20.04% over RNN with the second best MAPE value 13.72%. In Xi'an, the MAPE value of CERL for the 8-step SO_2 prediction is 18.39%, which improves 8.14% over ESN with the second best MAPE value 20.02%. The improvement is not bigger than previous experiments because that Pyrenn has the worst MAPE value 65.02% for the 8-step SO_2 prediction. Moreover, we can see that, although CERL has better performance, the MAPE values of CERL for air quality prediction are increased to the range of 6.59∼31.69%.

Discussions

The CERL improvements

To sum up, we can see that CERL provides better performance over the baseline models. In the 1-step prediction, all models have good performance to predict air quality. The average MAPE values of such models for all air pollutant factors (except O_3) fall in the range of 2.13∼6.05% and 2.56∼4.98%, respectively. This is because all such models are adequate for dealing with time serial data, especially for short-term prediction. Although CERL has superior performance over other models, the performance of such models is not obvious or even worse than that of the baseline models. As the step length increases, CERL has more obvious improvement, as shown in Tables 4 and 5. For example, the improvements of CERL in the 1-step, 3-step, 5-step, and 8-step prediction for $PM_{2.5}$ in Lanzhou are 1.82%, 8.01%, 9.98%, and 20.03%, respectively.

However, as the step length increases, the overall performance of all models declines. For example, the MAPE values of CERL 1-step, 3-step, and 5-step fall in the range of 2.01∼8.72%, 2.66∼11.84%, 5.05∼16.12%, respectively. As the step length is increased to 8, the MAPE values of CERL fall in the range of 6.59∼31.69%. We did not make further evaluation with bigger step length, since it makes not much sense if the prediction quality is worse than expected.

Diebold Mariano test

We further compare the performance of different models with a hypothesis testing method, called Diebold Mariano (DM) test. DM test is often used to check whether two forecasts for a time series are significantly different.

Let e_i^1 and e_i^2 (i=1,2) be the residuals for the two forecasts, i.e.,

$$e_i^1 = y_i - g_i \qquad e_i^2 = y_i - h_i \tag{10}$$

where y_i is actual value and g_i and h_i are predictive values of the two forecasts.

The loss function of two forecasts is defined as:

$$L(error_i^1) = (e_i^1)^2 \qquad L(error_i^2) = (e_i^2)^2 \tag{11}$$

The DM test statistic can be then defined by:

$$DM = \frac{\frac{1}{n}\sum_{i=1}^{n}(L(error_i^1) - L(error_i^2))}{\sqrt{\frac{S^2}{n}}}. \tag{12}$$

where S^2 is the an estimator of the variance of $d_i = L(error_i^1) - L(error_i^2)$. To check whether our CERL model is more accurate than other ones, we test the equal accuracy hypothesis. Given a significance level α, there are two hypotheses H_0 and H_1 defined as:

$$H_0 : L(error_i^1) = L(error_i^2) \tag{13}$$

$$H_1 : L(error_i^1) \neq L(error_i^2) \tag{14}$$

The null hypothesis H_0 denotes that there is no significant difference in the prediction performance of two forecasts. Against the null hypothesis H_0, the hypothesis

Table 5 The CERL MAPE improvements in different step prediction in Xi'an

	AQI	$PM_{2.5}$	PM_{10}	SO_2	NO_2	O_3	CO
1-step	2.68%	3.67%	2.93%	− 12.84%	− 1.19%	2.18%	1.69%
3-step	8.36%	11.35%	5.18%	11.58%	1.95%	2.28%	5.09%
5-step	21.45%	16.67%	12.65%	21.32%	17.32%	25.06%	10.92%
8-step	18.93%	17.14%	23.75%	8.15%	9.41%	18.36%	6.80%

Table 6 DM test of different models

City	Model	Forecast steps			
		1-step	3-step	5-step	8-step
Lanzhou	CFNN	− 2.44	− 2.17	− 3.14	− 3.82
	ESN	− 4.42	− 2.06	− 3.05	− 2.57
	RNN	− 4.49	− 2.16	− 2.58	− 4.04
	Pyrenn	− 3.54	− 5.76	− 17.43	− 9.59
Xi'an	CFNN	− 2.13	− 4.34	− 3.67	− 4.64
	ESN	− 3.57	− 2.76	− 3.18	− 2.16
	RNN	− 3.56	− 2.79	− 3.37	− 2.92
	Pyrenn	− 4.96	− 5.69	− 3.71	− 3.57

H_1 indicates that two forecasts have different levels of performance. The DM statistic follows approximately a standard normal distribution $N(0,1)$ under the null hypothesis. In this work, we set the significance level as 5%. In other words, the null hypothesis is rejected if $|DM| \leq 1.96$. Table 6 shows the DM test values for the $PM_{2.5}$ prediction between our CERL and other baseline models, i.e., CFNN, RNN, ESN, and Pyrenn. We can see that the lowest DM value is − 2.06. As a result, we can draw the conclusion that the null hypothesis is rejected and CERL has better performance than the other models.

To sum up, there are several reasons for the results. One is that CERL is a kind of ensemble model that employs different analytical models and then synthesizes their results into a single score in order to improve the prediction performance. Moreover, CERL not only involves of forward neural networks but also exploits the merits of recurrent neural networks that are designed for handling time serial data, such as RNN, ESN, and recurrent networks using previous outputs. In other words, CERL is able of capturing different underlying patterns in the data, thereby having superiority over the other baseline models.

Conclusion

This paper proposed a hybrid ensemble model CERL to exploit the merits of both forward neural networks and recurrent neural networks that are designed for handling time serial data to predict air quality hourly. Measured air pollutant factors including AQI, $PM_{2.5}$, PM_{10}, CO, SO_2, NO_2, and O_3 are used as input to predict air quality from 1 to 8 h ahead. Based on the air quality prediction in two rarely studied capital cities in Northwest of China, Lanzhou and Xi'an, CERL further improves the prediction performance over recurrent neural networks. However, this work is based on the prediction at the hour level, and does not have high accuracy for the long-term prediction. Our future study should be expanded to explore the air quality prediction at the day level. Moreover, this work only considers measured air pollutant factors and adding measured meteorological information into the air quality prediction may be another direction of our future research. In future, we will see how the information from multiple meteorological monitoring stations influences air quality prediction. In addition, we plan to employ convolutional neural network (CNN), a well-known deep learning model in air quality prediction since multiple factors are involved.

Funding information This work is supported by the National Natural Science Foundation of China (Project No. 61702240).

References

AirNow (2019) Air quality index (aqi) basics. https://airnow.gov/index.cfm?action=aqibasics.aqi. Accessed 29 Aug 2019

AirVisual (2019) Airvisual–air quality monitor and information you can trust. https://www.airvisual.com/. Accessed 26 Aug 2019

AirVisual I (2018) 2018 world air quality report-region & city pm2.5 ranking, Tech. Rep.

Atabay D (2019) pyrenn: a recurrent neural network toolbox for python and matlab–pyrenn 0.1 documentation. https://pyrenn.readthedocs.io/en/latest/. Accessed 02 Jul 2019

Athira V, Geetha P, Vinayakumar R, Soman KP (2018) Deepairnet: applying recurrent networks for air quality prediction. Procedia Computer Science 132:1394–1403. Retrieved from http://www.sciencedirect.com/science/article/pii/S1877050918308007 (International Conference on Computational Intelligence and Data Science)

Biancofiore F, Busilacchio M, Verdecchia M, Tomassetti B, Aruffo E, Bianco S, et al. (2017) Recursive neural network model for analysis and forecast of pm10 and pm2.5. Atmos Pollut Res 8(4):652–659. Retrieved from http://www.sciencedirect.com/science/article/pii/S1309104216304056

Cabaneros SM, Calautit JK, Hughes BR (2019) A review of artificial neural network models for ambient air pollution prediction. Environmental Modelling & Software 119:285–304. Retrieved from http://www.sciencedirect.com/science/article/pii/S1364815218306352

CNEMC (2019) China national environmental monitoring centre. http://www.cnemc.cn/. Accessed 08 Aug 2019

Feng X, Li Q, Zhu Y, Hou J, Jin L, Wang J (2015) Artificial neural networks forecasting of pm2.5 pollution using air mass trajectory based geographic model and wavelet transformation. Atmos Environ 107:118–128. Retrieved from http://www.sciencedirect.com/science/article/pii/S1352231015001491

Garcia JM, Teodoro F, Cerdeira R, Coelho LMR, Kumar P, Carvalho MG (2016) Developing a methodology to predict pm10 concentrations in urban areas using generalized linear models. Environ Technol 37(18):2316-2325. Retrieved from https://doi.org/10.1080/09593330.2016.1149228 (PMID: 26839052)

Genc DD, Yesilyurt C, Tuncel G (2010) Air pollution forecasting in Ankara, Turkey using air pollution index and its relation to assimilative capacity of the atmosphere. Environ Monit Assess 166(1):11–27. Retrieved from https://doi.org/10.1007/s10661-009-0981-y

He H, Li M, Wang W, Wang Z, Xue Y (2018) Prediction of pm2.5 concentration based on the similarity in air quality monitoring

network. Build and Environ 137:11–17. Retrieved from http://www.sciencedirect.com/science/article/pii/S0360132318301938

He Z, Chen Y, Shang Z, Li C, Li L, Xu M (2019) A novel wind speed forecasting model based on moving window and multi-objective particle swarm optimization algorithm. Appl Math Model 76:717–740. Retrieved from http://www.sciencedirect.com/science/article/pii/S0307904X19304020

HEI, IHME (2018) State of global air/2018 a special report on global exposure to air pollution and its disease burden (Tech. Rep.)

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. Retrieved from https://doi.org/10.1162/neco.1997.9.8.1735

Jaeger H (2001, 01) The"echo state" approach to analysing and training recurrent neural networks-with an erratum note'. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, 148

Jaeger H, Lukoševičius M, Popovici D, Siewert U (2007) Optimization and applications of echo state networks with leaky-integrator neurons. Neural Networks 20(3):335–352. Retrieved from http://www.sciencedirect.com/science/article/pii/S089360800700041X (Echo State Networks and Liquid State Machines)

Jiang F, He J, Tian T (2019) A clustering-based ensemble approach with improved pigeon-inspired optimization and extreme learning machine for air quality prediction. Appl Soft Comput 85:105827. Retrieved from http://www.sciencedirect.com/science/article/pii/S1568494619306088

Kwok L, Lam Y, Tam CY (2017) Developing a statistical based approach for predicting local air quality in complex terrain area. Atmos Pollut Res 8(1):114–126. Retrieved from http://www.sciencedirect.com/science/article/pii/S130910421630174X

Lang S, Bravo-Marquez F, Beckham C, Hall M, Frank E (2019) Wekadeeplearning4j: a deep learning package for weka based on deeplearning4j. Knowledge-Based Systems 178:48–50. Retrieved from http://www.sciencedirect.com/science/article/pii/S0950705119301789

Leksmono N, Longhurst J, Ling K, Chatterton T, Fisher B, Irwin J (2006) Assessment of the relationship between industrial and traffic sources contributing to air quality objective exceedences: a theoretical modelling exercise. Environmental Modelling & Software 21(4):494–500. Retrieved from http://www.sciencedirect.com/science/article/pii/S1364815204003123 (Urban Air Quality Modelling)

Li C, Hsu NC, Tsay SC (2011) A study on the potential applications of satellite data in air quality monitoring and forecasting. Atmos Environ 45(22):3663–3675. Retrieved from http://www.sciencedirect.com/science/article/pii/S1352231011004109

Lin YC, Lee SJ, Ouyang CS, Wu CH (2020) Air quality prediction by neuro-fuzzy modeling approach. Appl Soft Comput 86:105898. Retrieved from http://www.sciencedirect.com/science/article/pii/S1568494619306799

Lukoševičius M, Jaeger H (2009) Reservoir computing approaches to recurrent neural network training. Comput Sci Rev 3(3):127–149. Retrieved from http://www.sciencedirect.com/science/article/pii/S1574013709000173

Ma J, Cheng JC, Lin C, Tan Y, Zhang J (2019) Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques. Atmos Environ 214:116885. Retrieved from http://www.sciencedirect.com/science/article/pii/S1352231019305151

Ma J, Li Z, Cheng JC, Ding Y, Lin C, Xu Z (2019) Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network. Science of The Total Environment, pp. 135771. Retrieved from http://www.sciencedirect.com/science/article/pii/S0048969719357663

Shen L, Chen W, Wang J, Wei J, Yu Z (2016) Functional echo state network for time series classification. Inf Sci 373:1–20. Retrieved from http://www.sciencedirect.com/science/article/pii/S0020025516306661

Maciag PS, Kasabov N, Kryszkiewicz M, Bembenik R (2019) Air pollution prediction with clustering-based ensemble of evolving spiking neural networks and a case study for London area. Environmental Modelling & Software 118:262–280. Retrieved from http://www.sciencedirect.com/science/article/pii/S1364815218307448

Mallet V, Sportisse B (2008) Air quality modeling: from deterministic to stochastic approaches. Comput Math Appl 55(10):2329–2337. Retrieved from http://www.sciencedirect.com/science/article/pii/S089812210700733X (Advanced Numerical Algorithms for Large-Scale Computations)

Osowski S, Garanty K (2007) Forecasting of the daily meteorological pollution using wavelets and support vector machine. Eng Appl Artif Intel 20(6):745–755. Retrieved from http://www.sciencedirect.com/science/article/pii/S0952197606001904

Perez P, Reyes J (2006) An integrated neural network model for pm10 forecasting. Atmos Environ 40(16):2845–2851. Retrieved from http://www.sciencedirect.com/science/article/pii/S1352231006000495

Pineda F (1987) Generalization of back-propagation to recurrent neural networks. Phys Rev Lett 59(19):2229–2232

Shang Z, Deng T, He J, Duan X (2019) A novel model for hourly pm2.5 concentration prediction based on cart and eelm. Sci Total Environ 651:3043–3052. Retrieved from http://www.sciencedirect.com/science/article/pii/S0048969718340841

Singh KP, Gupta S, Kumar A, Shukla SP (2012) Linear and nonlinear modeling approaches for urban air quality prediction. Sci Total Environ 426:244–255. Retrieved from http://www.sciencedirect.com/science/article/pii/S0048969712004809

Stadlober E, Hörmann S, Pfeiler B (2008) Quality and performance of a pm10 daily forecasting model. Atmos Environ 42(6):1098–1109. Retrieved from http://www.sciencedirect.com/science/article/pii/S1352231007009909

Tengeleng S, Armand N (2014) Performance of using cascade forward back propagation neural networks for estimating rain parameters with rain drop size distribution. Atmosphere 5(2):454–472. Retrieved from https://www.mdpi.com/2073-4433/5/2/454

Wang J (2019) China air quality online monitoring and analysis platform. https://www.aqistudy.cn/. Accessed 14 Aug 2019

Wang J, Song G (2018) A deep spatial-temporal ensemble model for air quality prediction. Neurocomputing 314:198–206. Retrieved from http://www.sciencedirect.com/science/article/pii/S0925231218307859

Wang X (2019) Historical data of air quality in China. http://beijingair.sinaapp.com/. Accessed 04 Aug 2019

Warsito B, Santoso R, Suparti S, Yasin H (2018, 05) Cascade forward neural network for time series prediction. J Phys Conf Ser 1025:012097

Wu Q, Lin H (2019) A novel optimal-hybrid model for daily air quality index prediction considering air pollutant factors. Sci Total Environ 683:808–821. Retrieved from http://www.sciencedirect.com/science/article/pii/S0048969719323290

Deng M, Xu F, Wang H (2018) Prediction of hourly pm2.5 using a space-time support vector regression model. Atmos Environ 181:12–19. Retrieved from http://www.sciencedirect.com/science/article/pii/S1352231018301535

Yoon H, Jun SC, Hyun Y, Bae GO, Lee KK (2011) A comparative study of artificial neural networks and support vector machines for predicting groundwater levels in a coastal aquifer. J Hydrol

396(1):128–138. Retrieved from http://www.sciencedirect.com/science/article/pii/S0022169410006761

Zhang Y, Bocquet M, Mallet V, Seigneur C, Baklanov A (2012) Real-time air quality forecasting, part i: history, techniques, and current status. Atmos Environ 60:632–655. Retrieved from http://www.sciencedirect.com/science/article/pii/S1352231012005900

Zhao S, Yuan X, Xiao D, Zhang J, Li Z (2018) Airnet: a machine learning dataset for air quality forecasting. Retrieved from https://openreview.net/forum?id=SkymMAxAb