Aerosol and Air Quality Research, 19: 1400–1410, 2019
Copyright © Taiwan Association for Aerosol Research
ISSN: 1680-8584 print / 2071-1409 online
doi: 10.4209/aaqr.2018.12.0450

Evaluation of Different Machine Learning Approaches to Forecasting $PM_{2.5}$ Mass Concentrations

Hamed Karimian[1,2], Qi Li[2*], Chunlin Wu[2], Yanlin Qi[2], Yuqin Mo[2], Gong Chen[2,4], Xianfeng Zhang[2], Sonali Sachdeva[3]

[1] *School of Architecture, Surveying and Mapping Engineering, Jiangxi University of Science and Technology, Jiangxi 341000, China*
[2] *School of Earth and Space Science, Peking University, Beijing 100871, China*
[3] *Kavli Institute for Astronomy and Astrophysics, Peking University, Beijing 100871, China*
[4] *Laboratory of Ocean Environmental Big Data Analysis and Application, Shenzhen 518055, China*

ABSTRACT

With the rapid growth in the availability of data and computational technologies, multiple machine learning frameworks have been proposed for forecasting air pollution. However, the feasibility of these complex approaches has seldom been verified in developing countries, which generally suffer from heavy air pollution. To forecast $PM_{2.5}$ concentrations over different time intervals, we implemented three machine learning approaches: multiple additive regression trees (MART), a deep feedforward neural network (DFNN) and a new hybrid model based on long short-term memory (LSTM). By capturing temporal dependencies in the time series data, the LSTM model achieved the best results, with *RMSE* = 8.91 μg m^{-3} and *MAE* = 6.21 μg m^{-3}. It also explained 80% of the variability (R^2 = 0.8) in the $PM_{2.5}$ concentrations and predicted 75% of the pollution levels, proving that this methodology can be effective for forecasting and controlling air pollution.

Keywords: Air pollution; Machine learning; Neural networks; Deep learning; Prediction.

INTRODUCTION

As one of the major air pollutants, atmospheric aerosols are groups of solid or liquid particles suspended in the air and come from different sources and in various shapes and sizes. Moreover, the large portion of particulate matter is produced in the lowest layer of the atmosphere. In general, the finer the size of the particulate matter, the deeper it can penetrate inside the respiratory system where adsorption is more efficient. Particles reaching deep inside the lung are deposited by diffusion to the surface of alveoli, and water-soluble components can pass through cell membranes by simple passive diffusion. Previous works have investigated the linkage between exposure to fine particles (i.e., with diameters less than 2.5 μm ($PM_{2.5}$)) and premature mortality (Di *et al.*, 2017; Hung *et al.*, 2018). This highlights the importance of predicting air pollution concentrations so as to aid the generation of appropriate responses.

Methods for predicting air pollution concentrations can be broadly classified into two major categories: simulation-based and data mining-based methods. Simulation-based method incorporates physical (for generating meteorological and background parameters) and chemical models to simulate emission, transport and chemical transformation of air pollution (Grell *et al.*, 2005; Emmons *et al.*, 2010). However, this method suffers from numerical model uncertainties, and due to the lack of data, the parameterization of aerosol emissions is restricted (Karimian *et al.*, 2016). Data mining-based approach exploits statistical or machine learning techniques to detect patterns between predictors and dependent variables in the time series data. Linear mixed-effects regression (Wang *et al.*, 2017), multiple linear regression (Dimitriou, 2016), geographically weighted regression (Karimian *et al.*, 2017) and land use regression (Huang *et al.*, 2017) are some of the widely used models that have been developed by assuming a linear correlation between explanatory and response variables. However, sometimes the linear assumption in these models may not reflect the direct relation between the explanatory variables and the air pollutant concentrations (Huang *et al.*, 2017). In addition, aerosol optical depth (AOD) is one of the satellite-based products which has been used to produce surface distribution of $PM_{2.5}$. However, application of AOD for studies that aim to forecast hourly $PM_{2.5}$ concentrations

* Corresponding author.
Tel.: 13801378854
E-mail address: liqi@pku.edu.cn

is limited due to low temporal resolution and the issue of missing values (Qi *et al.*, 2019).

In recent years, machine learning methods have demonstrated their feasibility in nonlinear regimes. Thus, their applications in air pollution concentration forecasts are increasing (Peng, 2015). Gardner and Dorling (1998) presented the advantages of artificial neural networks (ANN) in dealing with nonlinear systems. Kukkonen *et al.* (2003) compared the performances of five ANN models, a linear model and a deterministic model to forecast NO_2 and PM_{10}. They concluded that the ANN-based models perform better, particularly in NO_2 forecasts. In addition to that, including meteorological data, especially planetary boundary layer height (PBL), can improve the accuracy of predictions (Hooyberghs *et al.*, 2005). Some recent works have proposed hybrid models and have claimed their robust performances in severe pollution scenarios (Feng *et al.*, 2015; Kumar, 2015; Tamas *et al.*, 2016; Perez and Menares, 2018).

Air pollution forecasts have attracted interest from scholars in Iran, as a region which suffers from notable air pollution problems. Kamali *et al.* (2015) proposed a hybrid model including the Kolmogorov–Zurbenko filter and an ANN to forecast PM_{10} concentrations over a station in Tehran. They claimed R^2 = 0.90 between predicted and observed values. Memarianfard and Hatami (2017) presented daily averaged $PM_{2.5}$ forecasts using a three-layer feedforward neural network (FNN) with daily averaged meteorological data (wind speed, relative humidity and temperature) obtained from a meteorological station. The performance of random forest feature selection in predictions of $PM_{2.5}$ was investigated by Shamsoddini *et al.* (2017). The authors claimed that better performance was achieved in comparison to linear regression model and FNN. Jamal and Nabizadeh Nodehi (2017) found improvement in $PM_{2.5}$ and air quality index forecasts ($RMSE$ = 21.26 μg m^{-3}) through a combination of decision trees and FNN. However, these predictions have been carried out using historical meteorological data observed from a station without considering the conditions at forecast time, which can have an influence on accuracy of the forecasts. Moreover, although multiple machine learning frameworks have been proposed for air pollution forecasts, it should be noted that most of these models have not been tested in developing countries with significantly higher $PM_{2.5}$ levels and different emission source profiles (Liu, 2013). To the best of our knowledge, considering the effect of temporal dependencies in air pollution data and deep neural networks (DNN) has not been investigated over the study area. In this study, we implement and evaluate three methods to forecast $PM_{2.5}$ concentrations, including a model based on machine learning (no neurons) and two models based on deep neural networks. Details of each of these techniques are presented in the following sections.

DATA AND METHODS

Study Area

Tehran, the capital of Iran (~51.1–51.6°E, 35.6–35.8°N), is located in arid region with a population of 13.3 million (native) plus 10 million (diurnal migration). It is one of the most polluted cities of Iran, where haze scenarios are reported frequently (Kamali *et al.*, 2015).

Ground Level $PM_{2.5}$

This study used hourly concentrations of $PM_{2.5}$ provided by Tehran Air Quality Control Company. This data was collected from 9 stations for a period of 4 years from 1 January 2013 to 31 December 2016 (Fig. 1). In order to improve the predictive performance of our models, we have omitted anomalies by performing an initial check. In addition to that, to account for the missing data due to instrumental malfunctions, we applied the interpolation method (before and after mean), as described in Ghasemifard *et al.* (2019). However, to ensure the accuracy of the process, time sequences with more than 3 hours of consecutive missing data were excluded from the interpolation and discarded from the dataset. This was done through partitioning the data in different blocks, where length of the blocks was based on the duration for which forecasting was desired. It was ensured that each block consists of required data. Our models have been trained using 60% of the dataset, and remaining 40% has been used for validation (20%) and testing (20%).

Meteorological Data

As described before, meteorological parameters are an important determinant of air pollution concentrations. Due to 700 m of altitude difference between the highest and lowest geographical points of Tehran, the weather conditions vary across the city (Habibi *et al.*, 2017). In consideration of that, improving on previous studies which used ground-based meteorological data from only one station, we collected hourly meteorological data, including temperature (T), PBL, surface-level pressure (P), east and north components of 10-m wind (U, V) and relative humidity (RH) from European Centre for Medium-Range Weather Forecasts (ECMWF) for the period of our study. This center provides high spatial resolution (0.1° × 0.1°) data with 10 days' forecast. A statistical summary and features of input data are given in Table 1. The values (maximum, minimum, mean and standard deviation) are based on hourly data of all stations. The mean value of $PM_{2.5}$ is 3 times higher than WHO guidelines for the annual average (10 μg m^{-3}). It has been found that usage of explanatory variables which are highly correlated reduces the applicability of models (Feng *et al.*, 2015; Karimian *et al.*, 2017). Variance inflation factor (VIF) detects the severity of redundancy by examining the correlation coefficient (R) between each pair of explanatory variables (Eq. (1)). A VIF lying between 5 and 10 indicates high correlation that may be problematic (Akinwande *et al.*, 2015). Table 2 shows the VIF values for different auxiliary variables. It can be seen that meteorological variables are weakly correlated and no redundancy is observed.

$$VIF = \frac{1}{1 - R^2} \tag{1}$$

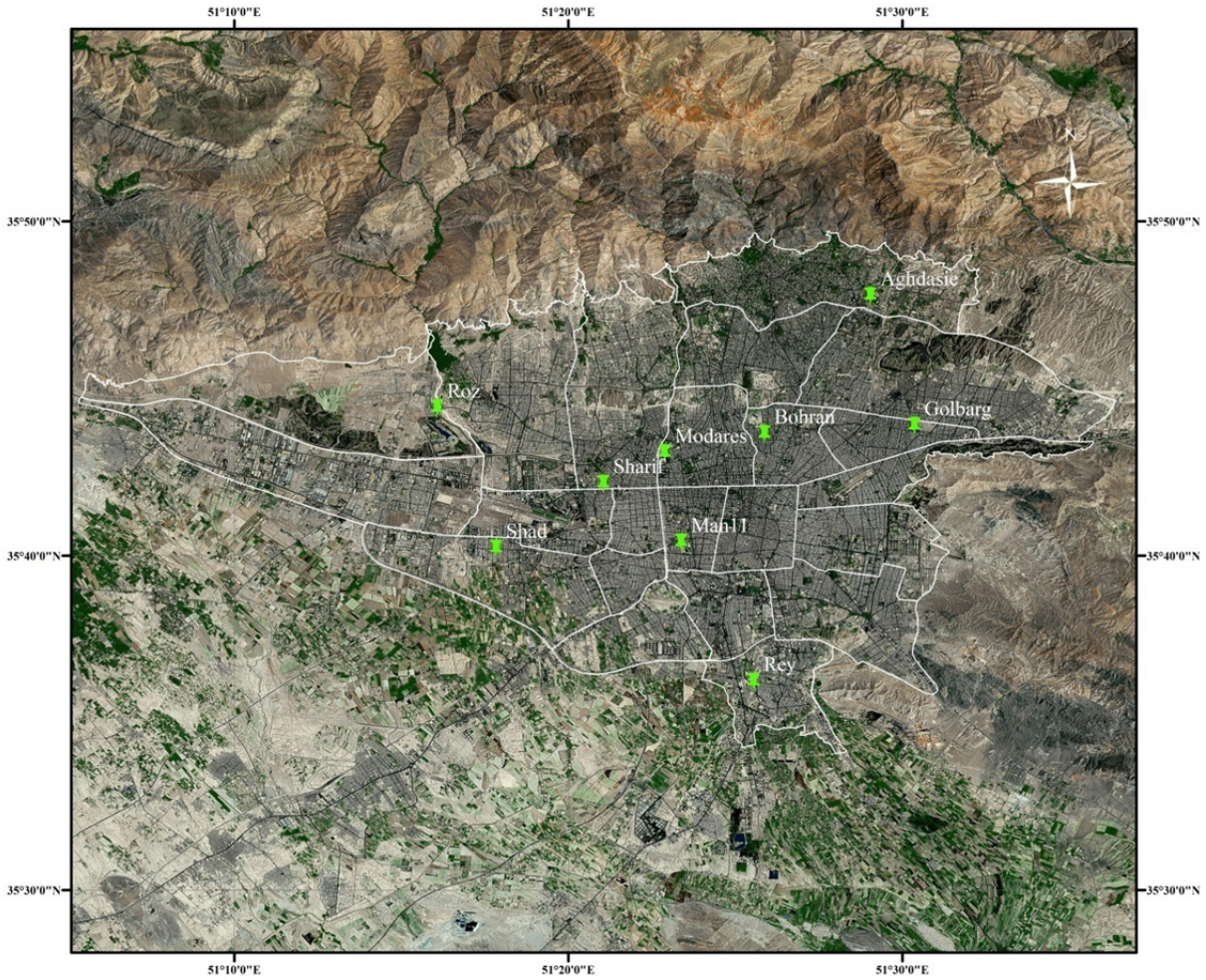

Fig. 1. Tehran metropolitan area and the air pollution monitoring stations (green pin).

Table 1. Statistical summary of input data from 1 January 2013 to 31 December 2016.

Variable	Unit	Range	Mean	Std. Dev.
$PM_{2.5}$	$\mu g\ m^{-3}$	[1, 300]	32	20
T	°C	[–23, 40]	13.50	11.6
PBL	m	[10, 4775]	620	864
P	Pa	[78,184, 86,321]	82124	1803
U	$m\ s^{-1}$	[4.88, 6.57]	0.04	1.12
V	$m\ s^{-1}$	[–4.55, 5.72]	–0.31	1.64
RH	%	[2, 100]	42	21

Table 2. VIF values between different explanatory variables.

	T	U	V	PBL	RH	P
T	-	1.004	1.639	1.471	2.702	1.002
U	1.004	-	1.006	1.191	1.008	1
V	1.639	1.006	-	2.326	1.351	1.008
PBL	1.471	1.191	2.326	-	1.370	1
RH	2.702	1.008	1.351	1.370	-	1
P	1.002	1	1.008	1	1	-

Multiple Additive Regression Trees

Multiple additive regression trees (MART) is a machine learning method proposed by Friedman (Friedman, 2002; Friedman and Meulman, 2003). It is an extension and improvement for classification and regression trees (Beriman *et al.*, 1984) that utilizes stochastic gradient boosting (SGB)

to convert a sequence of weak learners into a complex predictor. The idea of SGB came from bagging procedure (Breiman, 1996), in which the author claimed that better results could be obtained by injecting randomness into a model (Friedman, 2002). Therefore, at each iteration (M), a subsample of the training set is selected randomly, and a tree partitions the pseudo residuals ($\tilde{y}$) into J disjoint regions R_{JM} (Eq. (2)), where I is an indicator function (it is 1 if the condition is true and 0 otherwise). Consequently, the final model is composed of small trees ranging from hundreds to thousands, where each of them has brought an improvement to the overall model (Elish and Elish, 2009).

$$T_M\left(X;\{R_{JM}\}_1^J\right)=\sum_{J=1}^{J}\bar{y}_{JM}.I\left(x\in R_{JM}\right) \tag{2}$$

$$\bar{y}_{JM}=mean_{x_i\hat{I}R_{IM}}\left(\tilde{y}_{iM}\right) \tag{3}$$

$$\tilde{y}_{iM}=-\left[\frac{\partial L\left(y_i,F\left(x_i\right)\right)}{\partial F\left(x_i\right)}\right]_{F(x)=F_{m-1}(x)} \qquad i=1{:}n \tag{4}$$

Considering least-squares loss function ($L(y, F(x)) = (y - F(x))^2/2$), pseudo-residuals in Eq. (4) can simply be calculated as the difference between ground truth values and corresponding predicted ones. The initial value of F_{M-1} in the first step of algorithm (F_0) is defined as the average of the target variable in training dataset for the least-squares loss function.

$$\tilde{y}_{iM}=y_{iM}-F_{M-1}(x_i) \tag{5}$$

Following the modification proposed by Friedman and Meulman (2003) to conventional boosting, the current prediction, F_{M-1}, is then separately updated by randomly selected subsamples (instead of whole sample), which can be written as:

$$F_M(x)=F_{M-1}(x)+\alpha\sum_{j=1}^{J}\gamma_{iM}\,I(x\in R_{jM}) \tag{6}$$

$$\gamma_{jM}=argmin\sum_{x\in R_{jM}}L\left(y_i,F_{M-1}(x_i)+\gamma\right) \tag{7}$$

For least-squares loss function, γ_{jM} is the mean of current residuals in J^{th} terminal node. Hastie *et al.* (2009) demonstrated that $4 \le J \le 8$ works well for boosting. In this work, our trees have maximum of 6 terminal nodes. Considering the fact that slowing the learning stage leads to a model with better performance, the shrinkage factor, α ($0 < \alpha < 1$), regularizes the process and allows more (and different) trees to fit into the residuals. The optimal learning rate can be estimated through cross validation or a testing sample. However, it was found that small values, such as $\alpha = 0.1$, lead to the better results (Friedman and Meulman, 2003). Table 3 provides the details of our model. To forecast $PM_{2.5}$ concentrations, our model gets $PM_{2.5}$ concentration and meteorological data at current time (present) as well as forecasted meteorological data up to a desirable time as predictor variables (Table 1).

Table 3. MART set-up parameters.

Parameter	Value
Number of trees (iteration, M)	1000
Terminal nodes	6
Learning rate (α)	0.1
Loss function	Squared error
Number of predictors	79–151–295

Recurrent Neural Network

Artificial neural networks are a class of machine learning methods in which collections of connected units (neurons) enable the machine to learn patterns of different complex circumstances (responses) for their future predictions. Through proposing different structures of networks (units and functions), there is a large class of ANN models. Feedforward neural networks are one of the widely used ANN patterns which are formed by fully connected layers of neurons (at least three layers). Moreover, these connections follow the same direction, and there is no cycle or loop in their connectivity graph (Fan *et al.*, 2017). According to universal approximation theorem, an FNN with a single hidden layer can learn any function (Cybenko, 1989; Hornik *et al.*, 1989). However, to achieve this ability, the size of hidden layer may need to be unfeasibly large. Empirically, it was found that adding more layers to an FNN can improve the performance of the networks in different tasks (Goodfellow *et al.*, 2016). Therefore, in comparison with a normal FNN, deep neural networks contain more hidden layers or have different structures (connectivity between neurons) and learning methods.

Recurrent neural networks (RNN) are a class of DNN that, through applying cyclic (loop) connections, allow information to persist (a loop allows information to be passed from one step of the network to the next). Thus, they are specialized in time series processing. However, in practice, as the time sequence gets longer, the network forgets to train primary inputs. This issue is called "exploding" or "vanishing" gradients and arises due to the architecture of RNN (Bengio *et al.*, 1994). Considering a loss function, L, and a linear activation function, the gradient of L for the first hidden state with respect to the weight of input w_x can be written by chain rule as Eq. (8).

$$\frac{\partial L}{\partial w_x}=\left(\frac{\partial L}{\partial h_t}\right)\left(\prod_1^{t-1}w_h\right)\left(x_1\right) \tag{8}$$

In the above, w_x is the weight matrix that multiplies with input x_t in different time steps. It can be inferred that the overall gradient of loss function in the RNN (sum of the error gradient at each time step) contains the exponents of transposed weight matrix w_h, which is the weight multiplied against the hidden state. As a result, the gradient will explode

(if weights > 1.0) or shrink (if weights < 1.0) exponentially. This makes it difficult for RNN (i.e., it will diverge, be very slow or stop) to learn long dependencies. One solution to overcome this problem is to utilize the long short-term memory (LSTM) model architecture as a special class of RNN (Hochreiter and Schmidhuber, 1997).

Fig. 2 illustrates the schematic of an LSTM block at time step t, which we used in this study. It features four main elements, including the input gate (i), forget gate (f) and output gate (o). The fourth element, which is the key to an LSTM model, is the memory cell (cell state), which can be understood as a straight connection passing through a set of LSTM blocks with some minor linear interactions. Similar to the RNN structure, at each time step, the inputs of an LSTM block are input (x_t) and the hidden state output (h_{t-1}) from the prior time step block. In addition to that, the cell state (c_{t-1}) is the third input of a block. It is worth mentioning that the output of each gate is a vector with similar size to the hidden vector (h_t). The LSTM version implemented in this paper is similar to the one suggested by Graves (2013). However, to reduce the computational cost of network without malfunctioning (Greff *et al.*, 2017), peephole connections were removed.

$$f_t = \sigma(w_{xf}x_t + w_{hf}h_{t-1} + b_f) \quad (9)$$

$$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + b_i) \quad (10)$$

$$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + b_o) \quad (11)$$

$$pc = \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \quad (12)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot pc \quad (13)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (14)$$

As can be seen in Eqs. (9)–(11), for the forget, input and output gates, the nonlinearity is brought through the sigmoid activation function (σ). The output values of these gates range between 0 and 1, which allow them to control the flow of data inside and between LSTM blocks. For example, by looking at current inputs and through pointwise multiplication (Eq. (13)), the forget gate decides the portion of the data that should be removed from the previous memory cell. For potential values of memory cell (pc) and hidden state (Eqs. (12) and (14)), the tangent activation function (tan$h$) is used; its outputs are between –1 and 1, and its derivative can sustain for a long range before vanishing. The values of current cell memory (c_t) and hidden state (that are revealed outside a block) are calculated through element-wise multiplication. This multiplication is carried out against the output of forget gate (f_t varies in different time steps) rather than the repetitive matrix multiplication of weights in RNN (Eq. (8)). This can be inferred as one of the advantages of LSTM, as it avoids extreme vanishing or exploding of the gradients. It is also recommended to initialize the bias of the forget gate to 1 (Jozefowic *et al.*, 2015). These make LSTM-based models feasible to learn dependencies in long sequences and prevent vanishing of the gradients.

To forecast $PM_{2.5}$ concentrations over desirable time spans, we propose two types of DNN: One is based on deep FNN with three hidden layers (DFNN), and the other is a hybrid model comprising two LSTM layers and a DFNN with three hidden layers (Fig. 3). For simplicity, we call this model "LSTM" hereinafter. Our LSTM model gets the meteorological data (current (t_0) and hourly forecasted (t_1–t_n) as input for the two layer LSTM, in which the final output of the second layer is concatenated with current (t_0) $PM_{2.5}$ concentration and is treated as the input layer for the FNN. The final output of the FNN is forecasted $PM_{2.5}$ concentration at t_n. To train our models, mini batch root mean square prop (RMSprop) algorithm is employed. It is one of the optimization algorithms based on

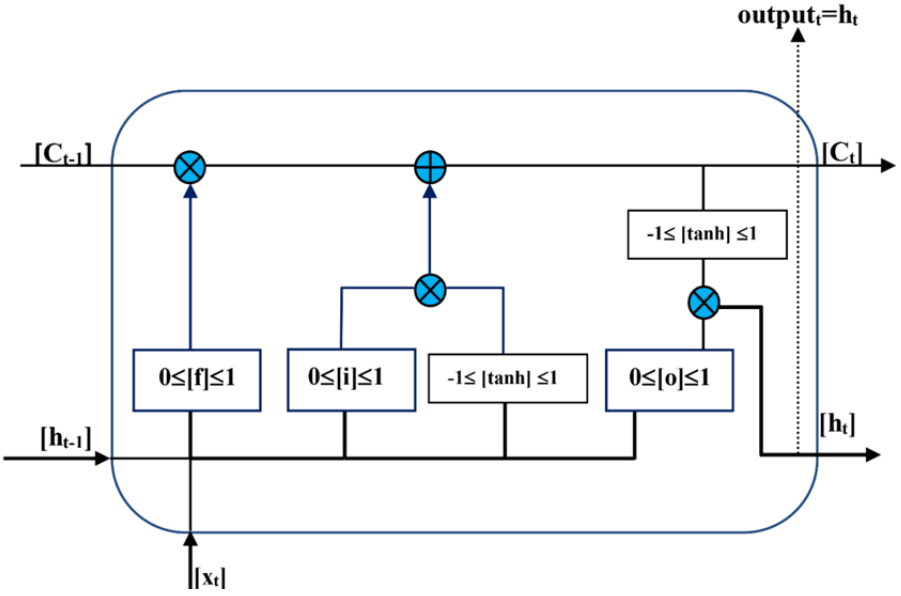

Fig. 2. Schematic of an LSTM block at time step t.

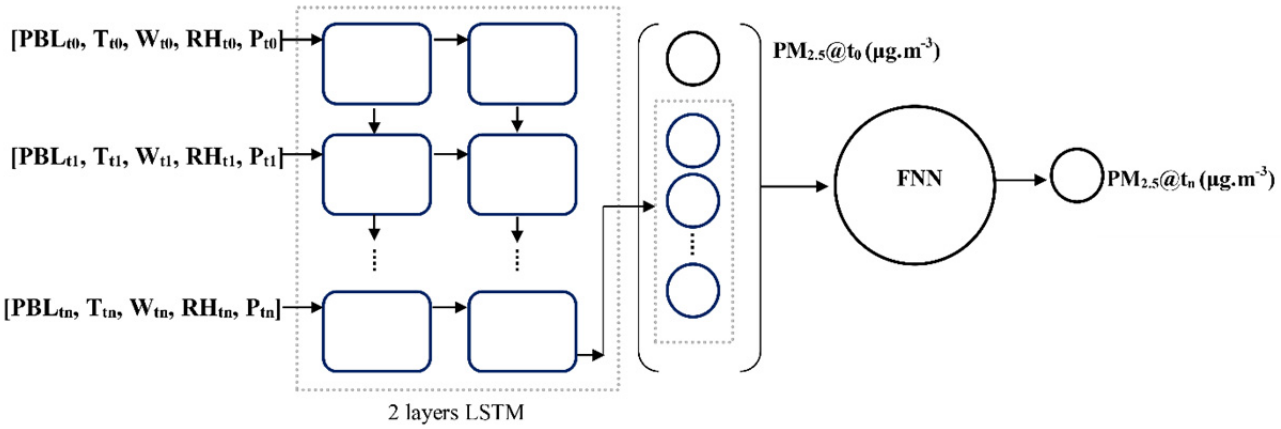

Fig. 3. Structure of the proposed LSTM model.

gradient descent method. The main idea of this algorithm is to speed up the process of gradient descent (Hinton *et al.*, 2012b). As a feature of most sophisticated models, overfitting, which refers to the acceptable performance of a model in training stage and failure over unobserved datasets, may occur. There are several techniques to avoid overfitting, which are known as *regularization* (Nielsen, 2015). We used the dropout technique, in which the architecture of a network is modified in each iteration by randomly removing hidden units and connections from a network. Through the dropout procedure, since a neuron cannot rely on the presence of other neurons, it is expected to learn more robust features. This improves the performance of different neural networks remarkably (Hinton *et al.*, 2012a). Early stopping is another technique that prevents overfitting. It stops training stage if the performance of a model on validation dataset fails to improve after an optional number of epochs (here, 100). The details of our proposed DNN models are provided in Table 4. The values of hyper-parameters in this table were selected through trial and error.

RESULTS AND DISCUSSIONS

To evaluate the performances of our proposed models, root mean square error (*RMSE*) and mean absolute error (*MAE*) were computed, which measure the closeness of the forecasts (F_i) to the observed values (O_i) over the test dataset. To analyze the prediction strength of different models, overall coefficient of determination (R^2) was derived as well. Note that these time intervals were selected for evaluation purpose and models are able to make forecasts over any time intervals.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(O_i - F_i)^2}{n}} \tag{15}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|O_i - F_i| \tag{16}$$

As can be seen in Table 5, the LSTM model showed better performance, with overall *RMSE* = 9.58 μg m^{-3}, than other models (*RMSE* = 19.53 and 13.32 μg m^{-3} for DFNN and MART, respectively) for different time intervals and over all stations. In contrast to DFNN and MART (where inputs from different times are all fed together to the models), these results demonstrate the importance of

Table 4. Features of our proposed DNN models.

Parameter	Value
LSTM layer	2
LSTM blocks	13–25–49
LSTM hidden neurons	64
Prediction length (hour)	12–24–48
FNN hidden layers (FNN/LSTM-FNN)	3
FNN hidden neurons	32
Mini batch size	512
Activation FNN (hidden-output)	Rectified linear unit (ReLU)
Loss function	Squared error
Number of iterations (epoch)	1000
Stopping point (epoch)	100
Dropout (%)	20
Optimizer	RMSprop

Table 5. Comparing the performance of three models in forecasting $PM_{2.5}$ concentrations within different time spans (12, 24 and 48 hours).

Station	*RMSE* (μg m^{-3})			*MAE* (μg m^{-3})		
	MART	DFNN	LSTM	MART	DFNN	LSTM
12 Hours						
Aghdasie	11.54	18.33	**8.94**	8.77	14.01	**6.58**
Roz	12.47	17.57	**9.03**	9.21	12.82	**6.77**
Golbarg	11.90	14.25	**8.80**	8.26	10.98	**6.56**
Bohran	16.21	21.12	**11.52**	10.88	16.12	**8.41**
Modares	12.09	16.55	**9.83**	8.59	12.34	**6.65**
Sharif	13.03	20.074	**10.98**	9.34	14.51	**7.71**
Man11	17.49	23.15	**11.73**	11.64	16.71	**8.35**
Shad	14.43	19.44	**11.40**	9.94	14.49	**7.82**
Rey	16.33	25.45	**10.69**	11.45	19.50	**7.91**
Overall	13.94	19.54	**10.32**	9.78	14.61	**7.41**
Overall R^2	0.50	0.43	**0.74**			
24 Hours						
Aghdasie	12.30	18.27	**8.23**	9.04	14.00	**5.80**
Roz	11.82	17.058	**8.86**	8.70	12.41	**6.33**
Golbarg	10.15	14.83	**7.82**	7.26	11.28	**5.69**
Bohran	13.41	21.13	**9.75**	9.58	16.02	**6.70**
Modares	12.32	16.55	**9.52**	8.56	12.30	**6.03**
Sharif	13.88	20.64	**10.46**	9.48	14.58	**6.93**
Man11	16.18	23.15	**10.85**	11.08	16.71	**7.49**
Shad	13.22	19.54	**10.02**	9.21	14.56	**6.96**
Rey	15.45	25.45	**9.30**	10.68	19.50	**6.76**
Overall	13.19	19.62	**9.42**	9.28	14.59	**6.52**
Overall R^2	0.53	0.43	**0.78**			
48 Hours						
Aghdasie	11.60	18.38	**7.54**	8.16	13.87	**5.59**
Roz	12.06	17.37	**7.69**	8.50	12.60	**5.68**
Golbarg	10.07	14.59	**7.03**	7.27	11.10	**5.30**
Bohran	15.38	21.26	**9.51**	10.12	16.28	**6.48**
Modares	10.83	16.55	**8.52**	7.71	12.30	**5.78**
Sharif	13.06	20.05	**10.14**	9.38	14.60	**6.48**
Man11	16.38	23.15	**10.61**	11.09	16.71	**7.34**
Shad	12.34	19.49	**10.52**	8.78	14.47	**7.02**
Rey	13.69	24.26	**8.68**	9.95	18.82	**6.23**
Overall	12.83	19.45	**8.91**	8.99	14.52	**6.21**
Overall R^2	0.56	0.49	**0.80**			

sequential feeding in LSTM. As an advanced machine learning method, our proposed MART model performed better than DFNN, and overall, it can explain over 50% of variability in $PM_{2.5}$ concentrations (R^2 = 0.53). We have discussed in Section 2.5 that to attain better results with an FNN, the hidden layer might be unfeasibly large. From our results, it can be inferred that sophisticated machine learning models (without neurons) such as MART give better results than a DFNN in $PM_{2.5}$ predictions. Increase in the length of forecasts leads to better performance of all three models studied here. This is especially seen in the case of our LSTM model (10.41 and 7.43 vs. 8.91 and 6.21 for 12- and 48-h predictions, respectively). Our understanding is that this improvement is caused by the fact that we also consider the meteorological conditions at the time of forecast. In general, models used historical data for training purposes; thus, accuracy of their predictions decreases as the length of prediction time gets longer (Qi *et al.*, 2019). This shows that our methodology is crucial for getting more accurate predictions. We will investigate this further with higher number of parameters involved. As shown in Table 5, the performances of models varied spatially. This may have been caused by data ($PM_{2.5}$) availability, which differs among stations. It highlights the role of other factors (e.g., emissions or land cover), which should be considered in future studies. Our LSTM model exhibited better performance than other studies, e.g., by Memarianfard and Hatami (2017) and Shamsoddini *et al.* (2017), which achieved R^2 = 0.30 for observed and daily predicted $PM_{2.5}$ and *RMSE* = 18.13 μg m^{-3} for daily averaged $PM_{2.5}$ predictions, respectively.

Fig. 4 shows the scatter plots of observed and the LSTM-forecasted $PM_{2.5}$ concentrations over 12-, 24- and 48-h intervals. There is a reasonable agreement between

ground truth and predicted values, and our LSTM model is able to explain 80% (R^2 = 0.8) of the variability in observed $PM_{2.5}$ concentrations. For different time intervals, the slopes of linear regression are less than 1, and intercepts are positive. This depicts the tendency of model to underestimate and overestimate high and low concentrations, respectively.

Since the ultimate goal of $PM_{2.5}$ forecasts in urban areas is to improve air quality and health, we validated the feasibility of our LSTM model (the most predictive model in this study) in air pollution level (APL) estimations. For this purpose, we sampled the data from 21 December 2014 to 27 December 2014. Note that this period was selected randomly and was not included in training or test datasets. The air quality index (AQI) and air pollution levels are reported based on the highest AQI derived from six major air pollutant concentrations. Table 6 illustrates the air quality and air pollution subindex level based on daily averaged $PM_{2.5}$ concentrations (Tamura and Tateishi, 1997). Fig. 5 illustrates the averaged surface distribution of 48-h predicted and observed $PM_{2.5}$ concentrations for the sample period using the inverse distance weighted interpolation as well as corresponding air pollution levels. It is worth mentioning that the $PM_{2.5}$ data for Modares station was missing for this period. Thus, this station is not shown in the figure. Generally speaking, the predicted distribution pattern follows the observed one, with *RMSE* = 10.32 μg m^{-3} and R^2 = 0.76. However, similar to the trend seen with the test dataset, for the stations with high concentrations, our predicted surface shows underestimation tendency. Except for two stations (Aghdasie and Sharif) with air pollution level unhealthy for sensitive groups, our model has the ability to estimate true APL. Consideration of more explanatory variables may give better results, especially for the stations where $PM_{2.5}$ concentrations exceed 140 (μg m^{-3}) in some of the hours during sample period (e.g., Sharif).

CONCLUSIONS

In recent years, researchers have been proposing advanced models for forecasting air pollutant concentrations. In this study, we evaluated three methods employed in $PM_{2.5}$ forecasting by implementing models based on machine learning (MART) and deep neural network (DFNN and LSTM) concepts. The results showed that the LSTM model, which obtained the lowest *RMSE* (8.91 μg m^{-3}) and *MAE* (6.21 μg m^{-3}) values in combination with the highest R^2 (0.8) value for 48-h predictions over the entire study area, outperformed the other two models. Additionally, the LSTM model accurately forecasted 75% of the air pollution levels based on the $PM_{2.5}$ concentrations, demonstrating the importance of sequential feeding in time series modeling. We also found that the advanced machine learning models, such as MART, produced better $PM_{2.5}$ estimates than the DFNN model.

In summary, the LSTM model was able to capture temporal dependencies in time series data, which increased the accuracy of its $PM_{2.5}$ forecasting. Therefore, this methodology can be used to predict the concentrations of different air pollutants. Furthermore, adding explanatory variables in the future will enhance this model's performance, opening the door to investigating new variables and methods.

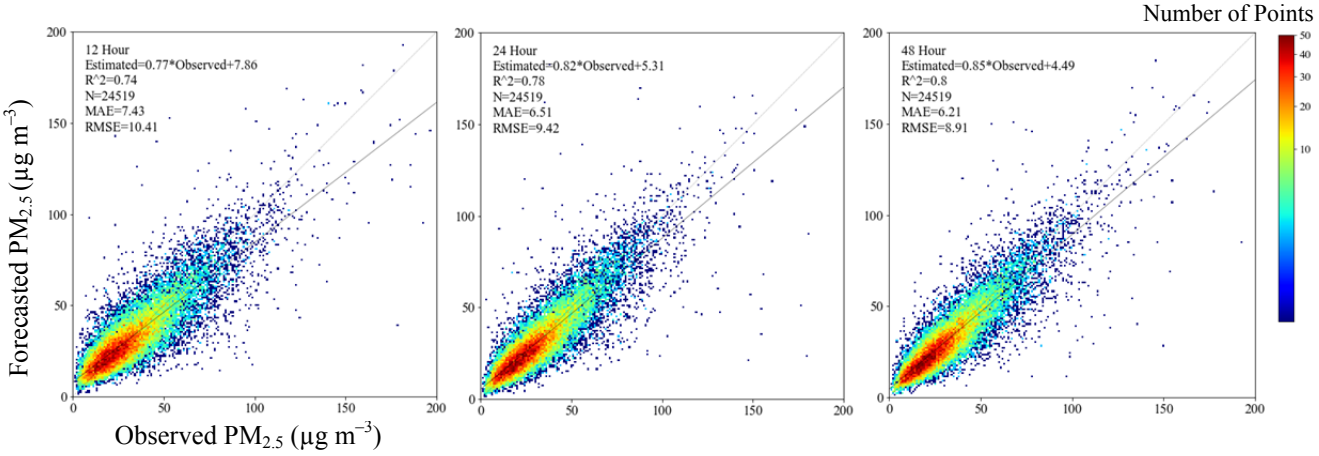

Fig. 4. Scatter plots of the observed and forecasted $PM_{2.5}$ concentrations by the LSTM model for 12-h (left), 24-h (middle) and 48-h (right) intervals.

Table 6. AQI and air pollution levels with corresponding daily averaged $PM_{2.5}$ concentrations.

AQI	Air pollution level	Max $PM_{2.5}$ concentration (μg m^{-3})
50	Good	15.4
100	Moderate	35
150	Lightly polluted (unhealthy for sensitive groups)	65.4
200	Moderately polluted (unhealthy)	150.4
300	Heavily polluted (very unhealthy)	250.4

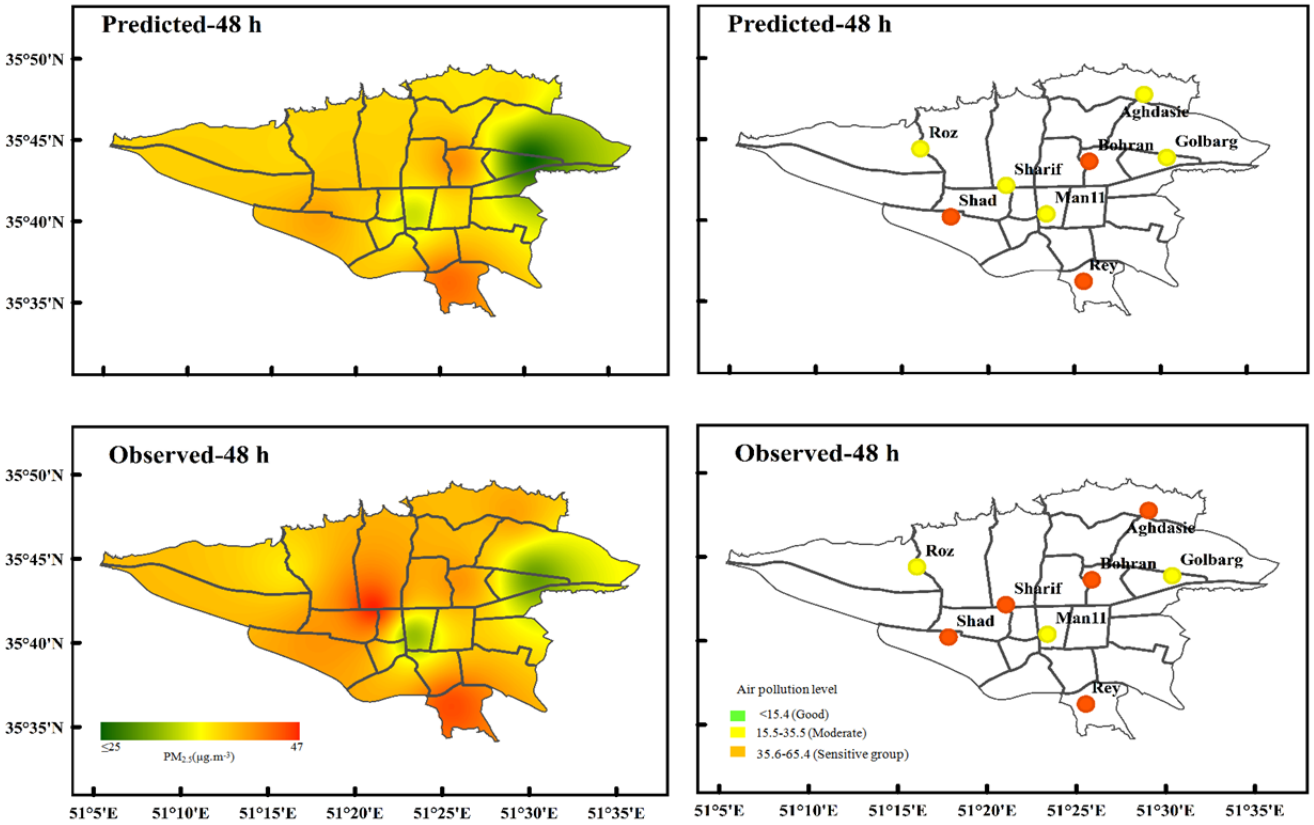

Fig. 5. Comparison between the forecasted (upper) and the observed (lower) surface distribution of $PM_{2.5}$. The correspondent air pollution levels are shown on the right.

ACKNOWLEDGMENTS

The study was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDA19030301).

REFERENCES

Akinwande, M.O., Dikko, H.G. and Samson, A. (2015). Variance inflation factor: As a condition for the inclusion of suppressor variable(s) in regression analysis. *Open J. Stat.* 5: 754–767.

Bengio, Y., Simard, P. and Frasconi, P. (1994). Learning long term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks* 5: 157–166.

Beriman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Calssification and regression tress*, Chapman and Hall.

Breiman, L. (1996). Bagging predictor. *Mach. Learn.* 26: 123–140.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* 2: 303–314.

Di, Q., Dai, L., Wang, Y., Zanobetti, A., Choirat, C., Schwartz, J.D. and Dominici, F. (2017). Association of short-term exposure to air pollution with mortality in older adults. *JAMA* 318: 2446–2456.

Dimitriou, K. (2016). Upgrading the estimation of daily PM_{10} concentrations utilizing prediction variables reflecting atmospheric processes. *Aerosol Air Qual. Res.* 16: 2245–2254.

Elish, M.O. and Elish, K.O. (2009). Application of treenet in predicting object-oriented software maintainability: A comparative study. 13th European Conference on Software Maintenance and Reengineering, pp. 69–78.

Emmons, L.K., Walters, S., Hess, P.G., Lamarque, J.F., Pfister, G.G., Fillmore, D., Granier, C., Guenther, A., Kinnison, D., Laepple, T., Orlando, J., Tie, X., Tyndall, G., Wiedinmyer, C., Baughcum, S.L. and Kloster, S. (2010). Description and evaluation of the Model for Ozone and Related chemical Tracers, version 4 (MOZART-4). *Geosci.Model Dev.* 3: 43–67.

Fan, J., Li, Q., Hou, J., Feng, X., Karimian, H. and Lin, S. (2017). A spatiotemporal prediction framework for air pollution based on deep RNN. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-4/W2: 15–22.

Feng, X., Li, Q., Zhu, Y., Hou, J., Jin, L. and Wang, J. (2015). Artificial neural networks forecasting of $PM_{2.5}$ pollution using air mass trajectory based geographic model and wavelet transformation. *Atmos. Environ.* 107: 118–128.

Friedman, J. (2002). Stochastic gradient boosting. *Comput. Stat. Data Anal.* 38: 367–378

Friedman, J.H. and Meulman, J.J. (2003). Multiple additive regression trees with application in epidemiology. *Stat. Med.* 22: 1365–1381.

Gardner, M.W. and Dorling, R.S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmos. Environ.* 32: 2627–2636.

Ghasemifard, H., Yuan, Y., Luepke, M., Schunk, C., Chen, J., Ries, L., Leuchner, M. and Menzel, A. (2018). Atmospheric CO_2 and $\delta^{13}C$ measurements from 2012 to 2014 at the environmental research station Schneefernerhaus, Germany: Technical Corrections, Temporal Variations and Trajectory Clustering. *Aerosol Air Qual. Res.* 19: 657–670

Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep learning, www.deeplearningbook.org.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint* arXiv:1308.0850.

Greff, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R. and Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Trans. Neural Networks Learn. Syst.* 28: 2222–2232.

Grell, G.A., Peckham, S.E., Schmitz, R., McKeen, S.A., Frost, G., Skamarock, W.C. and Eder, B. (2005). Fully coupled "online" chemistry within the WRF model. *Atmos. Environ.* 39: 6957–6975.

Habibi, R., Alesheikh, A.A., Mohammadinia, A. and Sharif, M. (2017). An assessment of spatial pattern characterization of air pollution: A case study of CO and $PM_{2.5}$ in Tehran, Iran. *ISPRS Int. J. Geo-Inf.* 6: 270.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The elements of statistical learning*, 2nd ed, Springer, New York, pp. 337–384.

Hinton, G., Srivastava, N. and Swersky, K. (2012b). Neural networks for machine learning Lecture 6a overview of mini-batch gradient descent.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.R. (2012a). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint* arXiv:1207.0580.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9: 1735–1780.

Hooyberghs, J., Mensink, C., Dumont, G., Fierens, F. and Brasseur, O. (2005). A neural network forecast for daily average PM concentrations in Belgium. *Atmos. Environ.* 39: 3279–3289.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* 2: 359–366.

Huang, L., Zhang, C. and Bi, J. (2017). Development of land use regression models for $PM_{2.5}$, SO_2, NO_2 and O3 in Nanjing, China. *Environ. Res.* 158: 542–552.

Hung, N.T., Ting, H.W. and Chi, K.H. (2018). Evaluation of the relative health risk impact of atmospheric PCDD/Fs in $PM_{2.5}$ in Taiwan *Aerosol Air Qual. Res.* 18: 2591–2599.

Jamal, A. and Nabizadeh Nodehi, R. (2017). Predicting air quality index based on meteorological data: A comparison of regression analysis, artificial neural networks and decision tree. *J. Air Pollut. Health* 2: 27–38.

Jozefowic, R., Zaremba, W. and Sutskever, I. (2015). An emprical exploration of recurrent network architectures. Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP, 37: 2342–2350.

Kamali, N., Zare Shahne, M. and Arhami, M. (2015). Implementing spectral decomposition of time series data in artificial neural networks to predict air pollutant concentrations. *Environ. Eng. Sci.* 32: 379–388.

Karimian, H., Li, Q., Li, C., Jin, L., Fan, J. and Li, Y. (2016). An improved method for monitoring fine particulate matter mass concentrations via satellite remote sensing. *Aerosol Air Qual. Res.* 16: 1081–1092.

Karimian, H., Li, Q., Li, C.C., Fan, J., Jin, L., Gong, C., Mo, Y., Hou, J. and Ahmad, A. (2017). Daily estimation of fine particulate matter mass concentration through satellite based aerosol optical depth. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-4/W2: 175–181.

Kukkonen, J., Partanen, L., Karppinen, A., Ruuskanen, J., Junninen, H., Kolehmainen, M., Niska, H., Dorling, S., Chatterton, T., Foxall, R. and Cawley, G. (2003). Extensive evaluation of neural network models for the prediction of NO_2 and PM_{10} concentrations, compared with a deterministic modelling system and measurements in central Helsinki. *Atmos. Environ.* 37: 4539–4550.

Kumar, U. (2015). An integrated SSA-ARIMA approach to make multiple day ahead forecasts for the daily maximum ambient O_3 concentration. *Aerosol Air Qual. Res.* 15: 208–219.

Liu, Y. (2013). New Directions: Satellite driven $PM_{2.5}$ exposure models to support targeted particle pollution health effects research. *Atmos. Environ.* 68: 52–53.

Memarianfard, M. and Hatami, A. (2017). Artificial neural network forecast application for fine particulate matter concentration using meteorological data. *Global J. Environ. Sci. Manage.* 3: 333–340.

Nielsen, A.M. (2015). *Neural networks and deep learning*, Determination Press.

Peng, H. (2015). *Air quality prediction by machine learning methods.* Thesis, University of British Colombia, UK.

Perez, P. and Menares, C. (2018). Forecasting of hourly $PM_{2.5}$ in south-west zone in Santiago de Chile. *Aerosol Air Qual. Res.* 18: 2666–2679.

Qi, Y., Li, Q., Karimian, H. and Liu, D. (2019). A hybrid model for spatiotemporal forecasting of $PM_{2.5}$ based on graph convolutional neural network and long short-term memory. *Sci. Total Environ.* 664: 1–10.

Shamsoddini, A., Aboodi, M.R. and Karami, J. (2017). Tehran air pollutants prediction based on random forest feature selection method. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-4/W4: 483–488.

Tamas, W., Notton, G., Paoli, C., Marie-Laure Nivet, M.L. and Voyant, C. (2016). Hybridization of air quality forecasting models using machine learning and clustering: An original approach to detect pollutant peaks. *Aerosol Air Qual. Res.* 16: 405–416.

Tamura, S. and Tateishi, M. (1997). Capabilities of a four-layered feedforward neural network:Four layers versus three. *IEEE Trans. Neural Networks* 8: 251–255.

Wang, W., Mao, F., Du, L., Pan, Z., Gong, W. and Fang, S. (2017). Deriving hourly $PM_{2.5}$ concentrations from Himawari-8 AODs over Beijing–Tianjin–Hebei in China. *Remote Sens.* 9: 858.

Received for review, January 12, 2019
Revised, April 1, 2019
Accepted, May 10, 2019

View publication stats