Deep learning method based on CMAQ for air pollutant prediction

Zhang Yun-fan, Luan Tian and Wang Guo-chang*

College of Economics, Jinan University, Guangzhou, China

Abstract

As air pollution becomes increasingly severe for most of developing countries, this situation causes many diseases, including respiratory disease, skin cancer and so on, which undermines the economic development and the health of the people. Many developing countries expend a great deal of human and material resources to control air pollution, but this task takes very long time to accomplish, and it is even difficult to reverse in some cases. An accurate prediction method is urgently needed to remind people to pay attention to certain air pollution issues in advance. The Community Multiscale Air Quality (CMAQ) approach is a widely used prediction method. However, its calculation cost is high, and it usually needs a supercomputer to obtain the CMAQ prediction. Moreover, the forecasting based on CMAQ is not very accurate. Many studies have recently used the long short-term memory(LSTM) and back-propagation(BP) neural networks to forecast air pollution based on

*Corresponding author:twanggc@jnu.edu.cn

meteorological data and the actual value of air contaminant concentration. However, the BP network is unstable long forecast intervals,whilst the LSTM network is not precise enough. Furthermore, these methods usually neglect the CMAQ data. To avoid these disadvantages, we utilise the generalised additive model (GAM) in this study and subsequently propose a GAM-LSTM-BP neural network model, hereafter referred to as the GLB model, based on the prediction of CMAQ, meteorological data and actual value of air contaminants concentration. From the real data analysis, the advantageous features of the proposed GLB model are as follows: (1) Aiming to sufficiently utilise CMAQ data, we use several lags of CMAQ data and the GAM model to automatically select the linear or nonlinear structure, thus avoiding the noise stack of the CMAQ. (2) We combine the LSTM and BP networks to provide a stable prediction model with a long interval (four days in advance) and a small error.

Keywords: deep learning; air pollutant; GAM; CMAQ; prediction

1 Introduction

With rapid economic development and the continuous advancement of industrialisation and urbanisation, air pollution has become increasingly severe for most countries, and this situation seriously hampers the developing economy and affects the health of the people. Many countries expend human and material resources to control air pollution, but the gains are not ideal. Hence, other ways need to be developed to control or decrease perniciousness of the air pollution. Many studies have focused on building precise predicted models, as advance and precise prediction can partially reduce the air pollution. For instance, high-polluting companies may be closed, or people can be warned to pay attention to this issue.

Traditional approaches have been used to predict the concentration of air contaminants in the atmosphere. One of the most popular models that give an efficient and economical depiction of aerosol dynamics in the atmosphere is the CMAQ model, which takes into account the regional tropospheric atmosphere as a whole, including the complex air conditions and relevant issue. The other ways of advanced air pollution prediction are the integration of the mesoscale meteorological model, pollution emission model and multiscale air quality model system for multiscale, multi-pollutant air quality forecasting, assessment and decision-making policy. The CMAQ model can predict most of the air pollution, such as $PM_{2.5}$, PM_{10}, SO_2, NO and NO_2, and the extinction of visible light by aerosols and cloud interactions with aerosols Binkowski and Roselle (2003) [1] Chen et al. (2013) [2]. The calculation cost of CMAQ is very high and usually needs a supercomputer to obtain the modelling result. Furthermore, the prediction given by the CMAQ model is not very precise. Other approaches can be used to predict the concentration of air contaminants based on meteorological data and original pollution data. For those models involving statistical methods, Meng (2009) [3] proposed the ARIMA model to forecast air pollution. Li et al. (2012) [4] analysed air pollutant concentrations in Wuwei City using the GM(1,1) model. Li et al. (2018) [5] also conducted a research on the prediction of air pollution index based on fractal manifold learning and the support vector machine method. Rubal and Kumar (2018) [6] proposed a new random forest method for air pollution prediction that combines state-of-the-art differential evolution strategies and the random forest method. For models involving deep learning technology, Sohn et al. (1999) [7] used the artificial neural network to study ozone prediction in different hours. Fan, Li and Hou (2017) [8] proposed three missing value fixing algorithms by using a technique for identifying the missing tag and missing interval, and then they combined the algorithms with the deep recurrent neural network. Their proposed method attained better performance compared with the deep feed forward neural networks and gradient boosting decision trees. OGM Khan et al . (2021) [9] proposed XNN combined with the long short-term memory (LSTM) method to forecast air pollution. Jin et al. (2021) [10] proposed a model that combines empirical mode decomposition LSTM to analyse air pollution data. Xiao et al. (2021) [11] studied the trend of several air contaminants and found that the performance of CMAQ prediction varies across different kinds of air contaminants.They also found that the CMAQ model usually entails a delay phenomenon, i.e. CMAQ usually reaches the peaks or valleys one or two days later compared with the real peaks.

The aforementioned statistical and neural network methods are very useful and have achieved remarkable application results. However, those methods do not consider the use of CMAQ data. Although CMAQ data can not fully offer an accurate prediction of air pollution, they still contain important information for predicting air contaminant concentration. In the present study, we combine CMAQ data, meteorological data and air contaminant concentration from a few days ago to predict future air contaminants concentration. However CMAQ prediction is not very precise, and using several days of CMAQ data can lead to noise stack and poor modelling performance. To solve this problem, we propose a method for selecting the important variables whilst deleting the unimportant variables. In paticular, we select the important variables via the generalised additive model (GAM) method. Compared with the variable selection methods based on linear models such as LASSO (Tibshiranni 1996) [12], SCAD (Fan and Li 2001) [13] and adaptive LASSO (Zou 2006) [14], our GAM-based method can be regarded as model-free and automatically select the linear and nonlinear structures.

Moreover, we use the neural network model and combine the GAM-selected CMAQ data to predict air pollution. Specifically, we propose the GAM-LSTM-BP

(GLB) model, which uses the LSTM and BP composite model. Firstly, we use LSTM to extract long-term sequence features, and secondly, we use BP to map and integrate the feature extraction of sequence data to construct the whole model.

2 Data and Predicted Model

2.1 Data collection and preprocessing

About 800 days (from 2018.5.23 to 2020.7.30) of data on air contaminant concentration, namely, NO_2, SO_2, CO, $PM_{2.5}$, PM_{10}, O_3, were collected from the RongGui monitoring stations in FoShan, GuangDong, China. The data are composed of auxiliary meteorological data, including daily minimum temperature, daily maximum temperature, daily average temperature, precipitation, daily maximum ten-minute mean wind direction (W_d), daily maximum ten-minute mean wind speed (W_v), daily maximum instantaneous wind direction (MW_d) and daily maximum instantaneous wind speed (MW_v). Subsequently, the air contaminant concentration was predicted based on the CMAQ model.

Aimed at avoiding slow convergence due to different data dimensions and prediction errors, the standard normalisation was applied to all data. As for missing data, the missing values were filled up by considering the average of data from nearby. Moreover, for further study, vectorisation was conducted on the daily maximum wind direction and daily maximum wind velocity, which are formulated as follows:

$$W_r = W_d \times \frac{\pi}{180}. \tag{1}$$

$$W_y = W_v \times \sin W_r. \tag{2}$$

$$W_x = W_v \times \cos W_r. \tag{3}$$

$$MW_r = MW_d \times \frac{\pi}{180}. \tag{4}$$

$$MW_y = MW_v \times \sin MW_r. \tag{5}$$

$$MW_x = MW_v \times \cos MW_r. \tag{6}$$

2.2 GAM-Based Feature Selection

Several days of CMAQ data were used to predict the air contaminants concentration. Thus, to avoid error stacking, we used a model-free feature selection method, based on GAM (Bakin,1999) [15]. The GAM-based method can automatically select the linear, nonlinear and zero functions i.e. the variable is a unimportant variable. The GAM-based variable selection method initially approximates the marginal nonparametric function $f_j(x_j)$ by using the cubic natural spline and then sperates the linear basis and nonlinear basis. The detailed procedure can be described as follows. Firstly, assume that the co-variate X and the response variable Y satisfy the GAM model.

$$Y = u + \sum_{j=1}^{p} f_j\left(X_j\right) + \varepsilon. \tag{7}$$

Denote the cubic natural spline as

$$\Phi_1\left(t\right) = 1, \Phi_2\left(t\right) = t, \Phi_{h+2}\left(t\right) = d_h\left(t\right) - d_{k_j-1}\left(t\right). \tag{8}$$

where $d_h(t) = \frac{(t-\tau_h)_+^3 - (t-\tau_{k_j})_+^3}{\tau_{k_j}-\tau_h}$,$k_j$ is the degree-of-freedom of the cubic natural spline and $\tau_1, \dots, \tau_{k_j}$ are the knots. Approximate the $f_j(X_j)$ by using the cubic natural spline as follows:

$$f\left(X_j\right) \approx \beta_1\Phi_1\left(X_j\right) + \beta_2\Phi_2\left(X_j\right) + \sum_{l=3}^{k_j}\beta_l\Phi_l(X_j). \tag{9}$$

For the sake of model identifiability, model (7) can be expressed as

$$Y = \sum_{j=1}^{p}\beta_{j2}\Phi_{12}\left(X_j\right) + \sum_{j=1}^{p}\sum_{l=3}^{k_j}\beta_{jl}\Phi_{h+2}\left(X_j\right) + \tilde{\varepsilon}. \tag{10}$$

To achieve the selection procedure, apply the Lasso-type penalty on the linear coefficient β_{j2} for $j = 1, \dots, p$ and the Group Lasso-type penalty on the nonlinear coefficient $\beta_{j3}, \dots, \beta_{jp}$ for $j = 1, \dots, p$. Subsequently, the coefficient can be estimated by

$$\begin{aligned}\left(\widehat{\beta_L}, \widehat{\beta_N}\right) \quad &= argmin \textstyle\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{p}X_{ij}\beta_{Lj} - \sum_{j=1}^{p}\sum_{l=3}^{k_j}\Phi_l(X_{ij})\beta_{jl}\right)^2 \\ &\qquad +\lambda_1\textstyle\sum_{j=1}^{p}|\beta_{Lj}| + \lambda_2\sum_{j=1}^{p}\sqrt{k_j-2}||\beta_{Nj}||_2.\end{aligned} \tag{11}$$

where $\widehat{\beta}_{Nj} = \left(\widehat{\beta}_{j3}, \widehat{\beta}_{j4}, \dots, \widehat{\beta}_{jk_j}\right)^T$, $\widehat{\beta}_N = \left(\widehat{\beta}_{N1}^T, \dots, \widehat{\beta}_{Np}^T\right)^T$, $\widehat{\beta}_L = (\widehat{\beta}_{12}, \dots, \widehat{\beta}_{p2})^T$, and $\lambda_{1,}$ $\lambda_2 > 0$ represents the tuning parameter, which can be selected by cross validation.

Consequently, the GAM-based variable selection method can successfully select the nonlinear structure if $\hat{\beta}_{Nj} \neq 0$ and automatically select the linear structure if $\hat{\beta}_{j2} \neq 0$ and $\hat{\beta}_{Nj} = 0$. Delete the variable from the model if $\hat{\beta}_{j2} = 0$ and $\hat{\beta}_{Nj} = 0$.

2.3 LSTM Neural Network

RNN has the problem of gradient descent, and an error criterion may be inadequate to train the parameters for tasks involving long-term dependencies (Bengio (1994) [16]). LSTM proposed by Hochreiter and Schmidhuber (1997) [17], aims to solve the defect of RNN. A traditional LSTM unit is composed of a cell, an input gate, a forget gate and an output gate. When a network for truly collecting and analysing the information in an arbitrarily long sequence needs to be established, LSTM introduces an internal state or cell state, which is the aforementioned cell c_t, to the neural network, allowing the network to process the transmission of gradient information, thus causing the capable of flow to be unchanged. The cell also provides a nonlinear information, namely the hidden state h. Here,c_t and h_t are formulated as follows:

$$c_t = f_t \circ c_{t-1} + i_t \circ \widetilde{c_t}. \tag{12}$$

$$h_t = o_t \circ \sigma_h(c_t). \tag{13}$$

where $\widetilde{c_t}$ is the cell input activation vector that can be formulated as

$$\widetilde{c_t} = \sigma_c(W_c x_t + U_c h_{t-1} + b_c). \tag{14}$$

The LSTM network also introduces a gating mechanism to regulate the transmission of the information.

1) The forget gate f_t controls over forgetting of information of the last iteration's internal state c_{t-1}.

2) The input gate i_t dominates the preservation of $\widetilde{c_t}$.

3) The output gate o_t regulates how much information in c_t shall be provided to h_t. These three gates are formulated as

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f). \tag{15}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i). \tag{16}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o). \tag{17}$$

- $x_t \in \mathbb{R}^d$: input vector to the LSTM unit
- $f_t \in (0,1)^h$: forget gate's activation vector
- $i_t \in (0,1)^h$: input/update gate's activation vector
- $o_t \in (0,1)^h$: output gate's activation vector
- $h_t \in (-1,1)^h$: hidden state vector, also known as the output vector of the LSTM unit
- $\widetilde{c_t} \in (-1,1)^h$: cell input activation vector
- $c_t \in \mathbb{R}^h$: cell state vector
- $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times h}$ *and* $b \in \mathbb{R}^h$: weight matrices and bias
- σ_g: $sigmoid\ function$
- σ_c: $hyperbolic\ tangent\ function$
- σ_h: $hyperbolic\ tangent\ function\ or$, as the peephole LSTM paper suggests,$\sigma_h(x) = x$.

On the basis of the abovementioned formulation, the LSTM can be implemented following the flowchart shown in Figure 1.

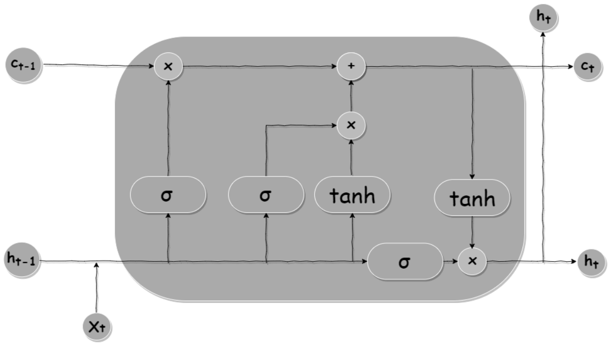

Figure 1: Demonstration of the LSTM network

2.3.1 BP Neural Network

The BP neural network (BPNN), as proposed by DE Rumelhart, GE Hinton, RJ Williams (1986) [18], contains three layers: the input layer (i.e., the preprocessed data comprising all selected features; the hidden layer providing a sufficient fitting process; and the output layer offering the prediction.

The basic unit of a neural network is the neuron divided into a linear unit $Z = A^0\beta$ or $Z = A^i\beta$, where A^i is the input of the i-th layer ($X = A^0$). Then, β is the weight of X in the layer and the nonlinear unit providing the output of this hidden layer $g\left(Z\right)$, where g is the activation function accomplishing the nonlinear calculation in the neural network in Figure.2

The fitting of the neural network relies on forward propagation and BP. Forward propagation completes the calculation from the input layer via the hidden layer to the output layer, which can be formulated as:

$$Z_i = A_i\beta. \tag{18}$$

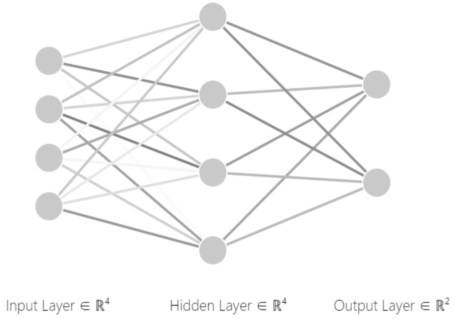

Figure 2: BPNN schematic diagram

$$A_{i+1} = g(Z_i). \tag{19}$$

Currently, activation function mostly includes, *sigmoid*, *tanh* and *ReLU*, which are all applied in our network. The formulation of *ReLU* is

$$g = \max(0, z). \tag{20}$$

The *ReLU* activation function has the advantage of a stable gradient when $z > 0$, ensuring efficiency of the gradient decent during training. Then, BP is applied to calculate the gradient of the parameters. The gradients are calculated based on the cross entropy loss of the BPNN, the gradients are calculated. However, the traditional gradient decent uses the same learning rate in the whole process, leading to inefficiency during training. Therefore, the Adam gradient decent algorithm was employed in this study. Adam gradient decent algorithm can be formulated as:

$$V_{d\beta} = \lambda_1 {*} V_{d\beta} + (1 - \lambda_1)\, d\beta; V_{du} = \lambda_1 {*} V_{du} + (1 - \lambda_1)\, du. \tag{21}$$

$$S_{d\beta} = \lambda_2 {*} S_{d\beta} + (1 - \lambda_2)\,(d\beta)^2\,; S_{du} = \lambda_2 {*} S_{du} + (1 - \lambda_2)\,(du)^2\,. \tag{22}$$

$$V_{d\beta}^{correct} = \frac{V_{d\beta}}{1 - {\lambda_1}^t}; V_{du}^{correct} = \frac{V_{du}}{1 - {\lambda_1}^t}. \tag{23}$$

$$S_{d\beta}^{correct} = \frac{S_{d\beta}}{1 - {\lambda_2}^t}; S_{du}^{correct} = \frac{S_{du}}{1 - {\lambda_2}^t}. \tag{24}$$

$$\beta = \beta - \alpha \frac{V_{d\beta}^{correct}}{\sqrt{S_{d\beta}^{correct} + \varepsilon}}; u = u - \alpha \frac{V_{du}^{correct}}{\sqrt{S_{du}^{correct} + \varepsilon}}. \tag{25}$$

where α,$\lambda_1$and λ_2 are hyperparameters, normally with values of $\lambda_1 = 0.9$ and $\lambda_2 = 0.999$, and t represents the iterations, with $\varepsilon = 1 \times 10^{-7}$.

2.4 Methodology Framework

Our proposed method comprises two parts: (1) feature selection (2) neural network analysis.

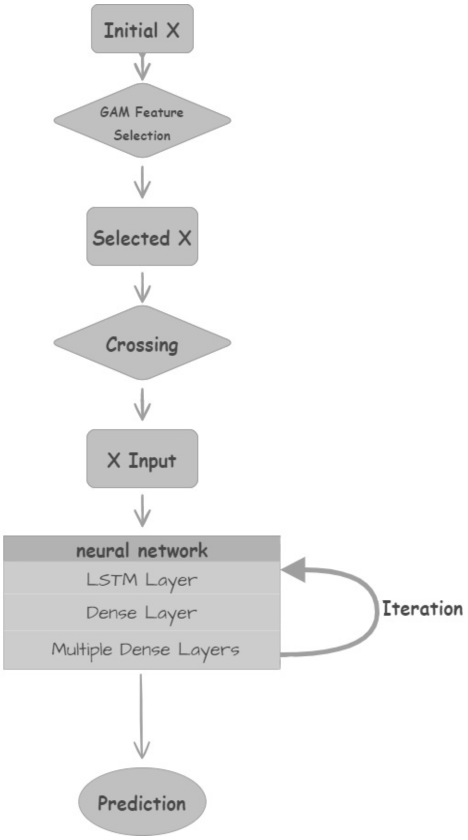

Figure 3: Model framework

As shown in Figure 3, for the first part, we implement the feature selection by using the GAM model, which is described in detail in Section 2.2. For the second part, the core of our methodology framework is to combine the LSTM and BP neural networks, allowing us to fully utilise their advantages whilst avoiding their disadvantages. Specifically, the LSTM layer is applied to extract long-term sequence features, and then the BP network is applied to analyse the extracted information from LSTM. However, even with the aforementioned feature selection, the overfitting problem still exists. To overcome this problem, we choose the most direct method, namely the dropout regularisation. Dropout regularisation randomly terminates some neurons in every iteration instead of applying penalty to the coefficients. On this basis, only part of the network is updated, in each iteration of the training process. Hence, the data can be analysed by a new network in each iteration. Thus, the final model is the combination of models of each iteration. The introduction of additional noise can enhance the robustness of the model.

3 Model Application and Result Analysis

We use data from 2018 to 2019 as the training dataset, and the data from 2020 as the testing sample. By implementing the proposed method, we are able to predict three air contaminant concentrations (NO_2, PM_{10} and CO) in the next four days. Before inputting all variables into the networks, GAM-based feature selection is applied to determine which CMAQ prediction of the prediction day should be utilised. Then, we consider three neural networks to train the data, namely, BP, LSTM and LSTM-BP. In the following section, the process of how the CMAQ prediction is selected and how to choose between those models are presented.

3.1 Feature Selection Based on GAM

Fu et al. (2020) [19] used air contaminants and meteorological factors to predict $PM_{2.5}$ by the ensemble empirical mode decomposition time series modeland found a strong interaction between $PM_{2.5}$ and other air contaminant concentrations. Air contaminant observations containing important information that should not be abandoned. In this study, we only apply feature selection to the CMAQ data. The results of NO_2 as an example are shown in Table 1. The selected feature varies across the $t + 2$, $t + 3$ and $t + 4$ models. Moreover, both $t + 3$ and $t + 4$ models select the

Table 1: Results of feature selection for NO_2.

CMAQ	β	t+2 model	t+3 model	t+4 model
$t+1$	β_L	0	-0.793	-0.128
	β_N	0	-0.645	0
	β_N	0	-0.603	0
	β_N	0	-0.354	0
$t+2$	β_L	3.222	0	0
	β_N	-0.023	0	0
	β_N	-0.051	0	0
	β_N	-0.078	0	0
$t+3$	β_L		3.454	0.022
	β_N		0	0.022
	β_N		0	0.025
	β_N		0	0.031
$t+4$	β_L			3.239
	β_N			0
	β_N			0
	β_N			0

CMAQ data of $t+1$ and $t+3$ as the important features, but the linear and nonlinear constructs present the opposite. By contrast, the $t+2$ model only selects the $t+2$ CMAQ data.

3.2 Forecast System

3.2.1 Model Training

Hyperparameter tuning has a significant impact on the training process of neural systems. Hence, many tests on the hyperparameter tuning of the modelwere implemented in this study. The rules can be summarised as follows:

1. LSTM should be selected as the first layer of all models, followed by BP.

2. All models should only contain a single layer of the LSTM network.

3. All models comprise at least three layers, but less than six layers of the BP layer. The layers of the BP layer increase as the forecast interval becomes longer.

4. The number of neurons in all LSTM layers and hidden layers (BP) is between 16 to 128.

5. All dropouts ranging from 0.5 to 0.75 should be only applied between the layers of the BP layer, as stipulated in most LSTM neural network studies stipulated.

Two evaluation measure criteria were used to evaluate the performance of the models, namely,

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2}, \tag{26}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{y_i} - y_i\right|. \tag{27}$$

3.3 Performance of the Three Considered Models

In this section, we initially compare the performance of the three considered models, namely, GAM-BP, GAM-LSTM and GLB for the three air contaminant concentrations (NO_2, PM_{10} and CO) in the next four days. Then we compare our proposed method with the CAMQ forecasting. The results are summarised in Tales 2-5, and Figures 4-15.

3.3.1 Results of NO_2

In this subsection, the results of NO_2 are summarised, and the detailed results are presented in Table 2 and Figures 4-7.

Table 2: Results of NO_2.

	GAM-BP		GAM-LSTM	
	$RMSE$	MAE	$RMSE$	MAE
$t+1$	13.954	10.498	13.868	10.862
$t+2$	16.028	12.757	20.559	18.728
$t+3$	18.187	13.332	21.324	19.500
$t+4$	20.905	15.826	23.755	21.046
	CMAQ		GLB	
	$RMSE$	MAE	$RMSE$	MAE
$t+1$	15.131	11.486	12.653	9.990
$t+2$	16.868	12.593	13.277	10.150
$t+3$	16.874	12.651	15.457	12.636
$t+4$	16.764	12.124	15.056	11.901

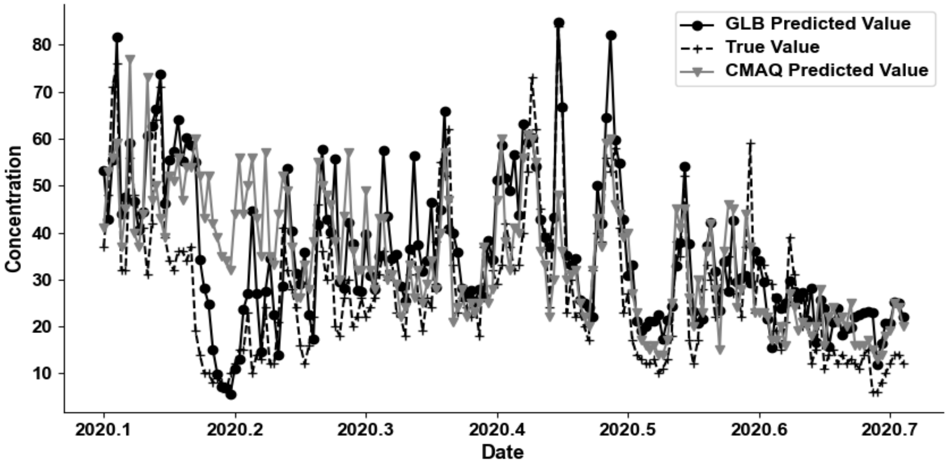

Figure 4: NO_2 t+1 prediction

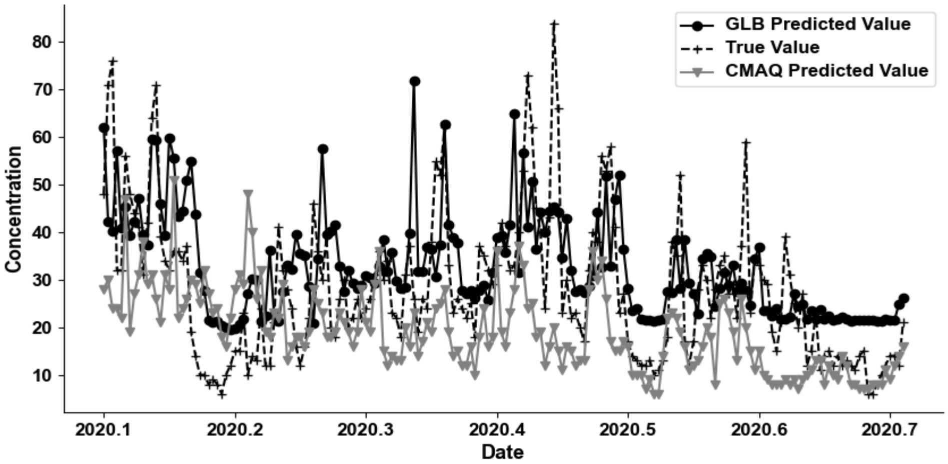

Figure 5: NO_2 t+2 prediction

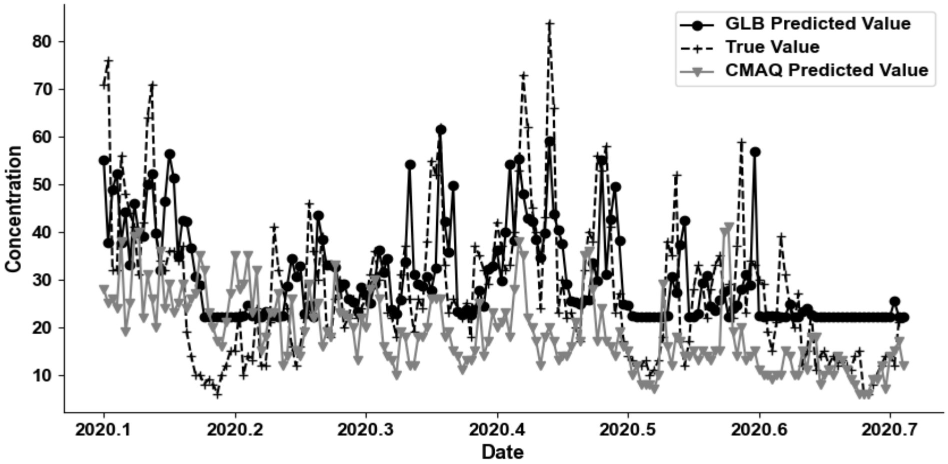

Figure 6: NO_2 t+3 prediction

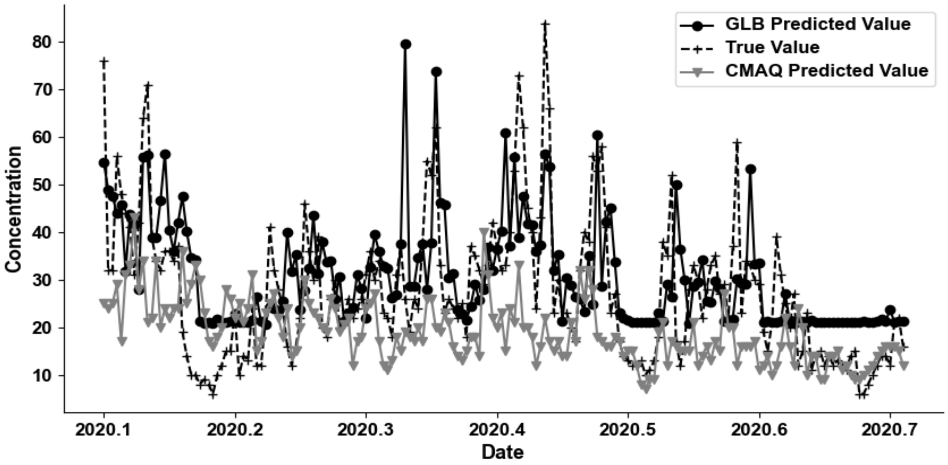

Figure 7: NO_2 t+4 prediction

The following findings can be derived from Table 2:

1) Compared with GAM-BP, GAM-LSTM and CMAQ, our proposed method (GLB) generally has the best performance, as shown by its lowest RMSE and MAE.

The differences between GLB and GAM-BP and between GLB and GAM-LSTM become more apparent as the forecasting days increase, whereas the difference between GLB and CMAQ decreases. The trends indicate that GLB performs better at the $t+1$ day, while it performs only slightly better than CMAQ at $t+2, t+3, t+4$ days.

2) GAM-BP has a better performance than CMAQ at the $t+1$ and $t+2$ forecasting days, but it performs worse than CMAQ at the $t+3$ and $t+4$ day. By contrast, GAM-LSTM only performs better than CMAQ at the $t+1$ day, but it performs worse than CMAQ on the other three days. GAM-BP usually has a better performance than GAM-LSTM.

The following findings can be derived from Figures 4-7.

3) CAMQ usually underestimates the true NO_2, especially at the $t+2$ and $t+3$ days. By contrast, our proposed GLB can overcome the problem and usually has more accurate results than CMAQ.

4) As our aim is to build an alarm system to remind people about air pollution, the peak of air contaminant concentrationmust be successfully predicted. Figures 4-7 show that our proposed method usually can more accurately predict the peak than CMAQ.

3.3.2 Results of PM_{10}

Similar to the calculation for NO_2, the results for PM_{10} are summarised. The details are shown in Table 3 and Figures 8-11.

Table 3: Results of PM_{10}.

	GAM-BP		GAM-LSTM	
	$RMSE$	MAE	$RMSE$	MAE
$t+1$	26.884	19.974	35.053	26.399
$t+2$	32.199	23.644	29.218	23.568
$t+3$	31.301	22.776	31.999	26.038
$t+4$	38.485	28.695	34.474	27.684
	CMAQ		GLB	
	$RMSE$	MAE	$RMSE$	MAE
$t+1$	22.323	17.764	19.844	14.101
$t+2$	26.598	21.354	20.596	14.310
$t+3$	26.630	20.434	20.846	13.883
$t+4$	26.232	19.723	21.304	15.148

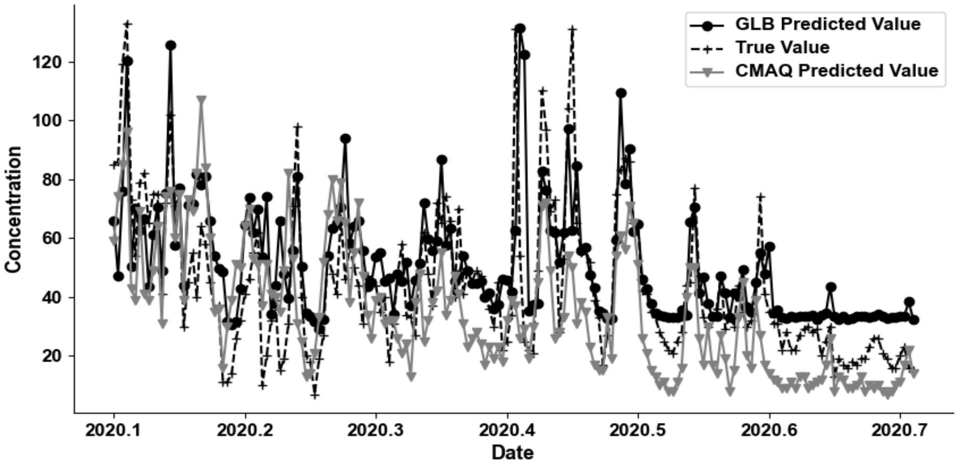

Figure 8: PM_{10} t+1 prediction

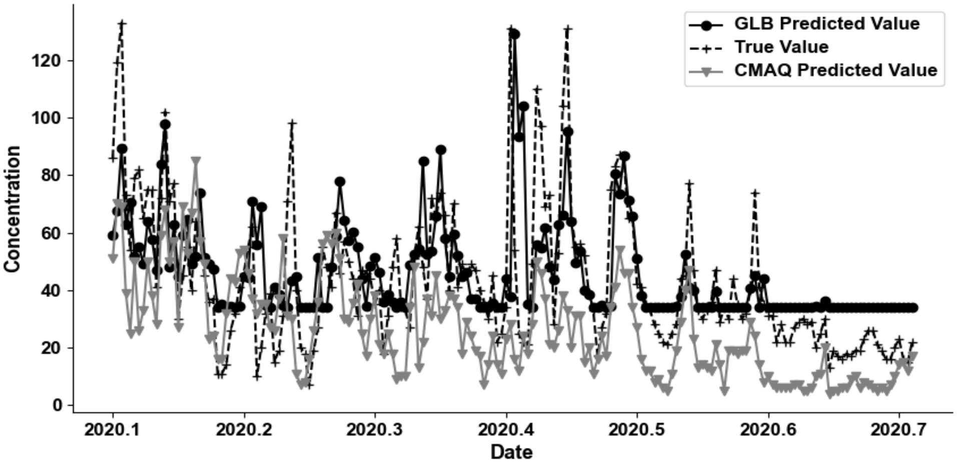

Figure 9: PM_{10} t+2 prediction

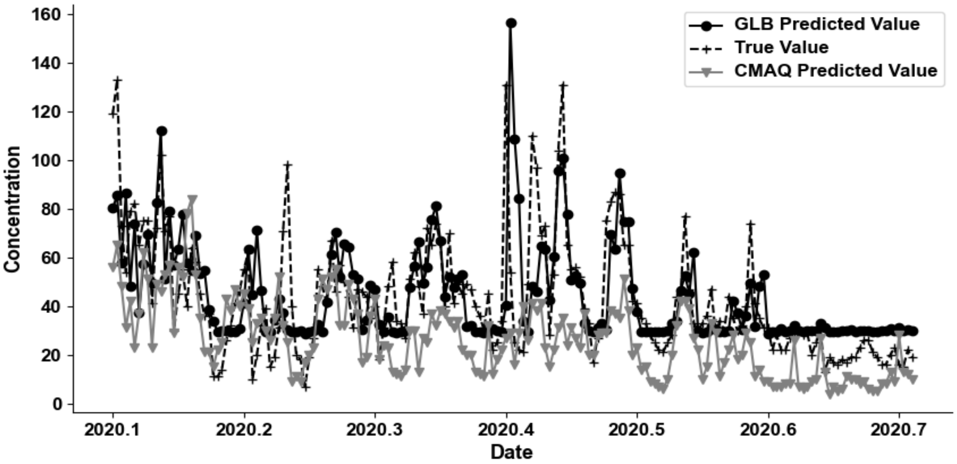

Figure 10: PM_{10} t+3 prediction

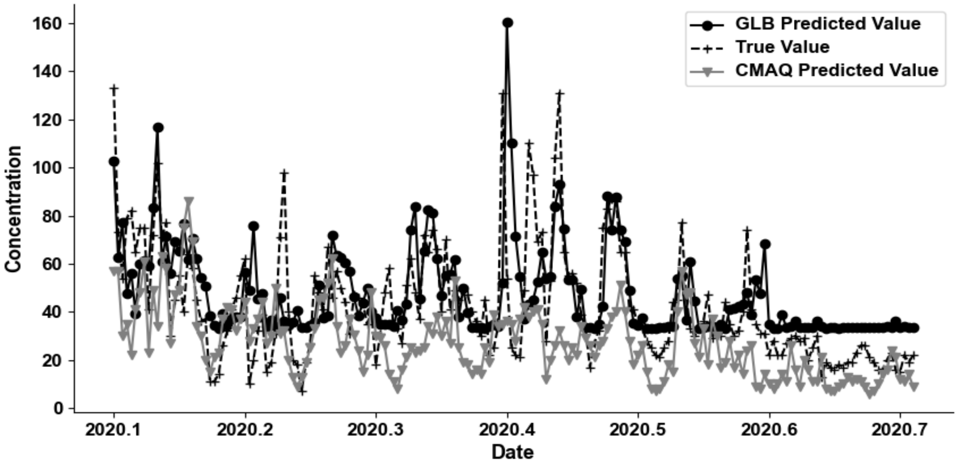

Figure 11: PM_{10} t+4 prediction

The following findings can be derived from Table 3:

1) Our proposed GLB offers the best forecasting technique amongst the four considered methods. The RMSE and MAE of GLM are at nearly three times less than those of CMAQ and nearly six times less than those of CMAQ. The difference between GLB and CMAQ increases as the prediction days increase, which indicates that GLB can be used for long-term prediction.

2) CMAQ performs better than GAM-BP and GAM-LSTM. Meanwhile GAM-BP and GAM-LSTM have similar performances.

The following findings can be derived from Figures 8-11:

1) Similar to NO_2, CMAQ usually underestimates PM_{10}, whereas our proposed GLB has good prediction results, although the forecast for June to July is somewhat overestimated.

2) Most of the peaks of PM_{10} appear in January and April. Our proposed GLB can precisely predict the peaks, whereas CMAQ usually underestimates them.

3.3.3 Results of CO

The results of CO are summarised in Table 4 and Figures 12-15.

Table 4: Result of CO.

	GAM-BP		GAM-LSTM	
	$RMSE$	MAE	$RMSE$	MAE
$t+1$	0.191	0.151	0.240	0.185
$t+2$	0.273	0.201	0.310	0.231
$t+3$	0.281	0.208	0.280	0.205
$t+4$	0.272	0.201	0.263	0.196
	CMAQ		GLB	
	$RMSE$	MAE	$RMSE$	MAE
$t+1$	0.439	0.426	0.147	0.121
$t+2$	0.529	0.518	0.184	0.147
$t+3$	0.515	0.510	0.178	0.146
$t+4$	0.500	0.493	0.181	0.145

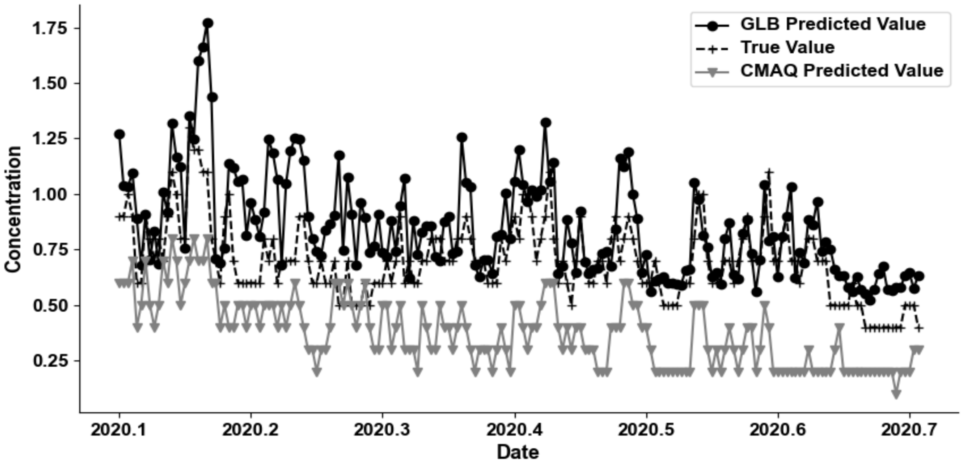

Figure 12: CO t+1 prediction

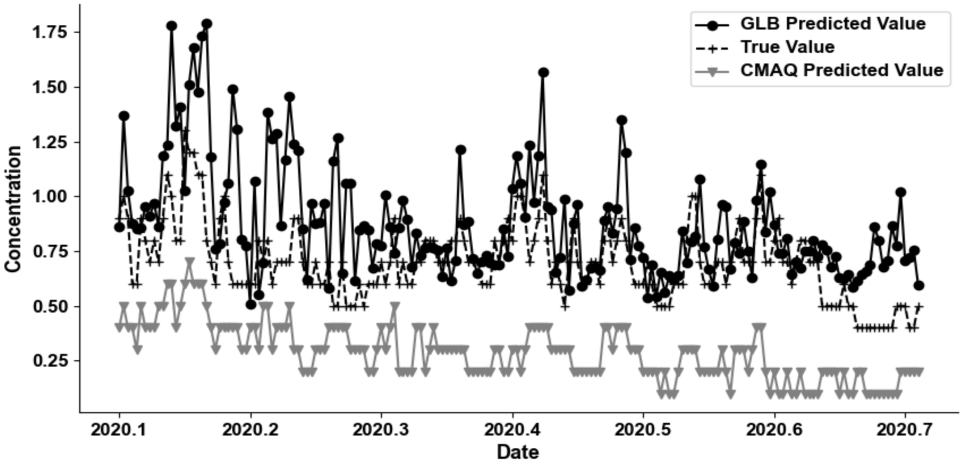

Figure 13: CO t+2 prediction

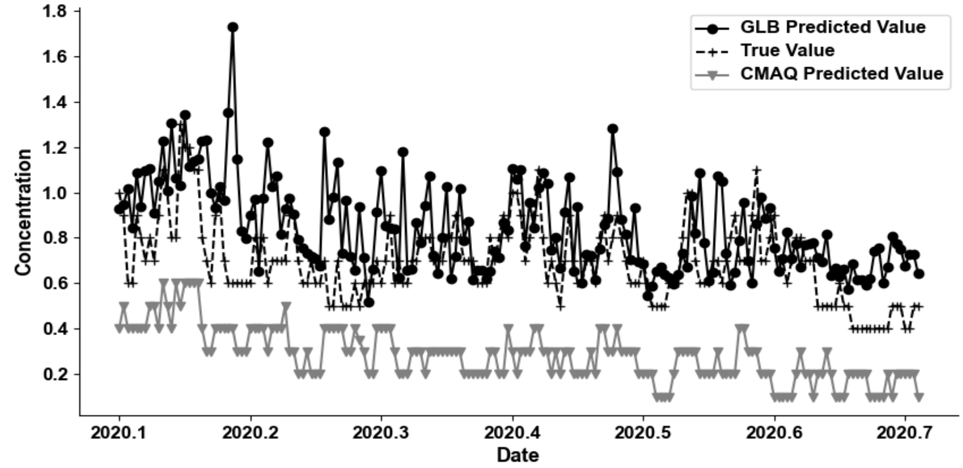

Figure 14: CO t+3 prediction

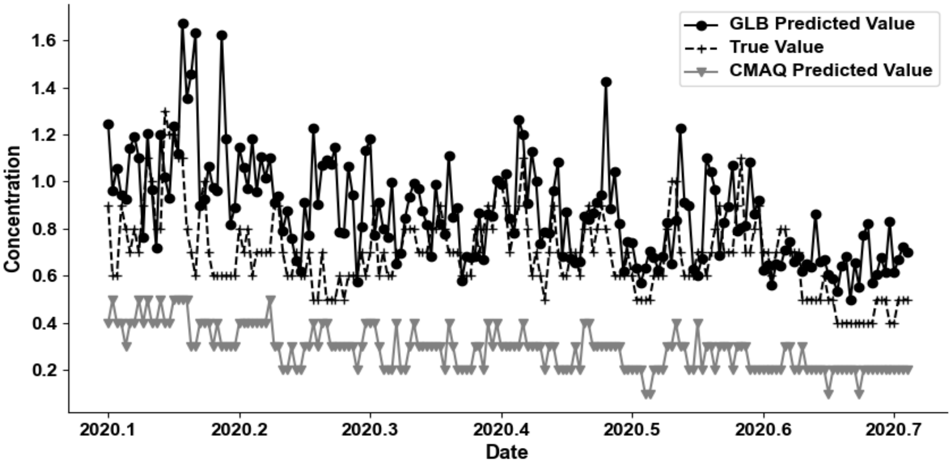

Figure 15: CO t+4 prediction

The following findings can be derived from Table 4:

1) The air contaminant concentration of CO is usually lower than NO_2 and PM_{10}; thus, its RMSE and MAE are much smaller than those of NO_2 and PM_{10}. Similar to the previous findings, our proposed GLB is superior amongst the four considered methods, as its RMSE and MAE are the lowest. The RMSE and MAE are stable for the predictors of $t+1,\ t+2,\ t+3,\ t+4$.

2) GAM-BP and GAM-LSTM both have better performances than GLB. The predictor of GAM-BP is slightly more accurate than that of GAM-LSTM. GAM-BP performs better than GAM-LSTM at the $t+1$ and $t+2$ days and its performance is similar to GAM-LSTM at the $t+3$ and $t+4$ days.

The following findings can be derived from Figures 12-15.

1) CMAQ clearly underestimates the CO, whereas our proposed GLB can give an accurate forecast, although it sometimes overestimate some of them.

2) For CMAQ, the underestimation in the forecast increases from $t+1$ to $t+4$.

Meanwhile, for GLB, the overestimation in forecast increases from $t + 1$ to $t + 4$.

4 Conclusion

This study investigates the process of how to improve air contaminant concentration forecast and proposes a new air contaminants concentration prediction model based on deep learning and CMAQ called the GLB (i.e., GAM-LSTM-BP) model. GLB comprises GAM-based feature selection, a LSTM neural network and a BP neural network.

Theoretically, GLB has the following advantages:

1. It combines traditional machine learning feature selection and neural networks, unlike most former studies relying on neural networks only.

2. Without any linear or nonlinear assumption, the GAM-based feature selection method can filter the features and autonomously identify the correlation between explained variable and independent variables whilst avoiding errors due to the false correlation assumption.

3. This neural network, which combines the LSTM layer and BP layers, can better extract information obtainable from historical contaminant concentration values, meteorological data and the prediction of CMAQ and GLB, whilst giving a more fitting prediction.

The results indicate the superiority of the proposed GLB. Firstly, it generally has a more outstanding performance in terms of RMSE and MAE compared with the three other models. For instance, the RMSE of GLB with respect to the $t + 2$ prediction of NO_2 decreases from 16.868 (RMSE of the t+2 prediction of CMAQ) to 13.277 (21.289%). Thus, CMAQ prediction can be amended, as expected in this study. Secondly, GLB can ensure the balance between RMSE and accuracy, offering an accurate air pollution warning with less false positives compared with the forecast of the three other models. Thirdly, GLB has the most stable prediction for the next four days, and it is more reliable than the three other models.

In conclusion, the proposed GLB can combine the idea of traditional machine learning feature selection and neural networks to revise and reuse the information obtainable from air contaminant concentration values, auxiliary meteorological data and CMAQ prediction. GLB has overcome some of the defects of the traditional single neural network model and the meteorological method of CMAQ,including long-term dependency or lack of precise forecast. This study has analysed three air contaminants. The results verify that the GLB Model perform well for each one of them, indicating its potential to be adopted for common air contaminant forecast. In addition, the collected data also comprise data of other monitoring stations. Whilst the distribution of air contaminants has a regional relationship, it was not considered in current research and thus can be addressed in future work.

5 Acknowledgments

The authors are very thankful to the editor, associate editor and reviewers for their constructive comments and suggestions, which have helped to significantly improve the paper.

References

[1] Binkowski F S, Roselle S J. Models'3 Community Multiscale Air Quality (CMAQ) model aerosol component 1. Model description[J]. Journal of geophysical research: Atmospheres, 2003, 108(D6).

[2] Xinhong Cheng, Xiangde Xu, Xingqin An et al. Inverse modeling of SO_2 and NO_x emissions using an adaptive nudging scheme implemented in CMAQ model in North China during heavy haze episodes in January 2013[J] Acta Scientiae Circumstantiae 2016,36(2):638-648. in Chinese

[3] Meng in 2009.7, Application of ARIMA Model in Air Pollution Index Forecast [J]. Statistics & Decision 2009, 2009(7): 33-35. in Chinese

[4] Li J , D Gong, Liu X . Prediction and Analysis of Air Pollutants Concentrations in Wuwei City of Gansu Province Based on GM(1,1) Model[J]. Environmental Science and Management, 2012, 27(1):17-29.

[5] Li et al. in 2018.3 Air Pollution Index Prediction Model of SVM Based on Fractal Manifold Learning [J]. Journal of Systems Science and Mathematical Sciences, 2018, 38(11): 1296-1306. In Chinese

[6] Rubal, D Kumar. Evolving Differential evolution method with random forest for prediction of Air Pollution[J]. Procedia Computer Science, 2018, 132:824-833.

[7] Sohn S H, Oh S C, Yeo Y K. Prediction of air pollutants by using an artificial neural network[J]. Korean Journal of Chemical Engineering, 1999, 16(3): 382-387.

[8] Fan J, Li Q, Hou J, et al. A spatiotemporal prediction framework for air pollution based on deep RNN[J]. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2017, 4: 15.

[9] Khan O G M, Youssef A, El-Saadany E, et al. LSTM-based approach to detect cyber attacks on market-based congestion management methods[C] IEEE Power & Energy Society General Meeting (PESGM). IEEE, 2021: 1-5.

[10] Jin et al. in 2021.4, Prediction of Outlet SO2 Concentration Based on Variable Selection and EMD-LSTM Network[J] Proceedings of the CSEE,2021,41(24) in Chinese

[11] XIAO Delin;DENG Shihuai;DENG Xiaohan et al. Analysis of Ambient Air Quality Variation Trend and CMAQ Model Forecast System in Urban Areas of Dazhou City[J]. Environmental Monitoring in China, 2021, 37(04):92-103 in Chinese

[12] Tibshirani R. Regression shrinkage and selection via the lasso[J]. Journal of the Royal Statistical Society: Series B, 1996, 58(1): 267-288.

[13] Jianqing Fan & Runze Li (2001) Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties, Journal of the American Statistical Association, 96:456, 1348-1360

[14] Hui Zou (2006) The Adaptive Lasso and Its Oracle Properties, Journal of the American Statistical Association, 101:476, 1418-1429

[15] Bakin S. Adaptive regression and model selection in data mining problems[D]: PhD Thesis. Australian National University, Canberra. 1999.

[16] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE transactions on neural networks, 1994, 5(2): 157-166.

[17] Hochreiter S , Schmidhuber J . Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.

[18] DE Rumelhart, Hinton G E , Williams R J . Learning Representations by Back Propagating Errors[J]. Nature, 1986, 323(6088):533-536.

[19] Fu H, Zhang Y, Liao C, et al. Investigating $PM_{2.5}$ responses to other air pollutants and meteorological factors across multiple temporal scales[J]. Scientific reports, 2020, 10(1): 1-10.