CrossMark

RESEARCH ARTICLE

Deep learning architecture for air quality predictions

Xiang Li[1,3] · Ling Peng[1] (ID) · Yuan Hu[1,3] · Jing Shao[2] · Tianhe Chi[1]

Abstract With the rapid development of urbanization and industrialization, many developing countries are suffering from heavy air pollution. Governments and citizens have expressed increasing concern regarding air pollution because it affects human health and sustainable development worldwide. Current air quality prediction methods mainly use shallow models; however, these methods produce unsatisfactory results, which inspired us to investigate methods of predicting air quality based on deep architecture models. In this paper, a novel spatiotemporal deep learning (STDL)-based air quality prediction method that inherently considers spatial and temporal correlations is proposed. A stacked autoencoder (SAE) model is used to extract inherent air quality features, and it is trained in a greedy layer-wise manner. Compared with traditional time series prediction models, our model can predict the air quality of all stations simultaneously and shows the temporal stability in all seasons. Moreover, a comparison with the spatiotemporal artificial neural network (STANN), auto regression moving average (ARMA), and support vector regression (SVR) models demonstrates that the proposed method of performing air quality predictions has a superior performance.

✉ Ling Peng
pengling@radi.ac.cn

[1] Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China

[2] Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China

[3] University of Chinese Academy of Sciences, Beijing 100049, China

Introduction

Air pollution is a serious environmental issue that is attracting increasing attention globally (Kurt and Oktay 2010). Many developing countries suffer from heavy air pollution. For example, extreme air pollution events have frequently occurred in China in recent years, especially in the Beijing, Tianjin, and Hubei districts. According to Reports on the State of the Environment in China (2015), among 338 monitored cities, 265 (78.4 %) were below the national healthy air quality standard, and the percentage of days below the standard reached 23.3 % on average.

Particulate matter with an aerodynamic diameter of or less than 2.5 μm (PM2.5) represents an air pollutant that can be inhaled via nasal passages to the throat and even the lungs. Long-term exposure to PM2.5 increases the incidence of associated diseases (e.g., respiratory and cardiovascular diseases, reduced lung function, and heart attacks) in humans (Künzli et al. 2000; Bravo and Bell 2011). Obtaining real-time air quality information is of great importance for air pollution control and for protecting humans from adverse health impacts due to air pollution (Zheng et al. 2013). Hence, it is necessary to conduct air quality prediction to better reflect the changing trend of air pollution and to provide prompt and complete environmental quality information for environmental management decisions, as well as to avoid serious air pollution accidents (Chen et al. 2013).

Many studies have focused on air quality predictions, and the following two types of methods are generally used: deterministic and statistical. A deterministic method employs theoretical meteorological emissions and chemical models (Bruckman 1993; Coats 1996; Guocai 2004; Jeong et al.

2011) to simulate pollutant discharge, its transfer and diffusion processes, and removal processes using dynamic data of a limited number of monitoring stations in a model-driven way (Kim et al. 2010; Baklanov et al. 2008). Representative methods, such as CMAQ (Chen et al. 2014) and WRF-Chem (Saide et al. 2011), are widely used for urban air quality forecasting. However, due to unreliable pollutant emission data, complicated underlying surface conditions, and an incomplete theoretical foundation, the simulation results suffer from low prediction accuracy (Vautard et al. 2007; Stern et al. 2008).

However, compared with these complicated theoretical models, statistical methods simply use a statistical modeling technique to predict the air quality in a data-driven manner. Straightforward methods such as the multiple linear regression (MLR) (Li et al. 2011) model and the auto regression moving average (ARMA) (Box and Jenkins 1970) model are commonly used for air quality prediction. However, these methods usually yield limited accuracy due to their inability to model nonlinear patterns; thus, they cannot predict extreme air pollutant concentrations (Goyal et al. 2006). A promising alternative to these linear models are artificial neural networks (ANNs) (Gardner and Dorling 1998; Hooyberghs et al. 2005; Lal and Tripathy 2012; Sánchez et al. 2013) and support vector regression (SVR) models (Nieto et al. 2013; Suárez Sánchez et al. 2013; Hájek and Olej 2012). A previous study showed that an ANN model was more accurate than the linear models (such as ARMA or MLR) because the air quality data presented clearer nonlinear patterns than linear patterns (Prybutok et al. 2000). Studies have also used a combination of these models for air quality predictions, and the results have shown that hybrid methods have a better predictive performance than single models (Díaz-Robles et al. 2008; Chen et al. 2013; Sánchez et al. 2013).

However, all these methods usually predict air quality at each station separately and neglect the high spatial correlations between stations. Spatial correlations generally occur between environmental variables (Legendre 1993). Air quality for all monitoring stations was highly correlated, thereby reflecting air pollutant dispersion patterns to some extent (Jerrett et al. 2005; Kracht et al. 2015). Therefore, it is important to fully model spatiotemporal correlations for air quality predictions.

Spatiotemporal prediction models that include air quality as a spatiotemporal process have been introduced, such as the spatiotemporal auto regression moving average (STARMA) (Martin and Oeppen 1975), the spatiotemporal artificial neural network (STANN) (Nguyen et al. 2012), and the spatiotemporal support vector regression (STSVR) models (Cheng et al. 2007). These methods can deal with nonlinear spatiotemporal features to a certain degree; however, the shallow models commonly use hand engineering to extract low level features. Thus, the performance of these models is greatly affected by artificial features, which inspired us to examine the air quality prediction problem in terms of deep architecture models

capable of capturing these spatiotemporal features for accurate predictions.

Recently, deep learning, a new potential machine learning methodology, has attracted considerable academic and industrial attention (Bengio 2009) and has been successfully applied to image classification, natural language processing, prediction task, object detection, artificial intelligence, motion modeling, etc. (Silver et al. 2016; Hinton et al. 2006; Zhang et al. 2015; Collobert and Weston 2008; Mohamed et al. 2011; Bengio 2009; Chan et al. 2015). Deep learning algorithms use multiple-layer architectures to extract the inherent features of data layer-by-layer from the lowest to the highest level, and they can identify representative structure in data. Because air quality process is inherently complicated, its temporal trends and spatial distribution are affected by various factors, such as air pollutant emissions and deposition, weather conditions, traffic flow, human activities, and so on. This situation has increased the difficulty of using traditional shallow models, especially for providing a good representation of air quality features. Deep learning algorithms can extract representative air quality features without prior knowledge and may lead to a good performance for air quality predictions.

In this paper, we introduced a deep learning-based method for air quality predictions. A stacked autoencoder model is used to extract representative spatiotemporal air quality features, and it is trained in a greedy layer-wise manner. Thus, spatial and temporal correlations are inherently considered in the model. Furthermore, experimental results have demonstrated that the proposed method for air quality predictions has superior performance.

The main novelty and contributions of this paper are summarized as follows:

1. We introduced the deep learning approach for research on air quality prediction. The latent air quality features can be automatically learned using a stacked autoencoder model, and the learned representations are used to construct a regression model for air quality prediction.
2. We treated the regional air quality as a spatiotemporal process and used the deep learning algorithm to build a spatiotemporal prediction framework, which considers the spatial and temporal correlations of air quality data in the modeling process. The experimental results demonstrated the advantages of this approach over time series models.
3. Our model can predict the air quality of all monitoring stations simultaneously and shows a satisfactory seasonal stability.

The remainder of this paper is structured as follows: the "Methodology" section presents the deep learning-based approach for air quality predictions; the "Experiment and

results" section discusses the experiments and results; and the "Conclusion" section presents the concluding remarks.

Methodology

First, a stacked autoencoder model is introduced. The stacked autoencoder model is a widely used deep learning architecture that incorporates autoencoders as building blocks to construct a deep network (Bengio et al. 2007).

Autoencoder

An autoencoder is a neural network that attempts to reconstruct its inputs (Lv et al. 2015). To accomplish this reconstruction and obtain a good representation, the autoencoder must capture the most important features of the input using methods that include principle component analysis (PCA). Figure 1 illustrates a basic schema of an autoencoder. Given a set of training samples $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(N)}\}$ in which $x^{(i)} \in R^d$, an autoencoder first encodes the input vector x to a higher-level hidden representation y based on equation (1), and then it decodes the representation y back to a reconstruction z, calculated as in equation (2):

$$y = f(W_1 x + b) \tag{1}$$

$$z = g(W_2 y + c) \tag{2}$$

where W_1 and W_2 are weight matrixes and b and c are bias vectors. We employed the logistic sigmoid function $1/(1 + \exp.(-x))$ for $f(x)$ and $g(x)$ in this study. The parameters of this neural network are optimized to minimize the average reconstruction error,

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} L\left(x^{(i)}, z^{(i)}\right). \tag{3}$$

Here, L is a loss function. We used the traditional squared error in our model.

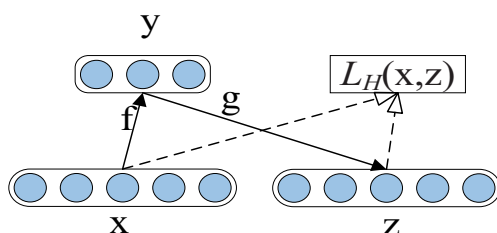However, the reconstruction criterion alone cannot guarantee the extraction of representative features because

Fig. 1 Autoencoder architecture. The autoencoder transforms input vector x to y via the encoder f and attempts to reconstruct x via the decoder g to produce reconstruction z. The reconstruction error is measured by the loss $L_H(\mathrm{x},\mathrm{z})$

it can lead to the straightforward solution of "simply copy the input" or similarly undesirable solutions that maximize mutual information (Vincent et al. 2010). To force the autoencoder to extract more robust features and prevent it from simply learning the identity, Ranzato introduced the sparse over-complete (i.e., higher dimension than the input) representation method (Poultney et al. 2006; Boureau and Cun 2008). A sparse over-complete representation can be perceived as a compressed representation because it has implicit compressibility due to the large amounts of deactivated hidden units rather than an explicit lower dimensionality (Vincent et al. 2008, 2010). To achieve the sparse representation, a sparsity restraint is embedded into the reconstruction error:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left\| x^{(i)}, z^{(i)} \right\|^2 + \lambda \left(\|W_1\|^2 + \|W_2\|^2 \right) \tag{4}$$

$$+ \mu \sum_{j=1}^{H_D} \mathrm{KL}\left(\rho \| \rho_j\right).$$

where $\|W_1\|^2$ and $\|W_2\|^2$ are the regulation terms, $\mathrm{KL}(\rho\|\rho_j)$ is the sparsity term, λ and μ are the weights for the regulation term and sparsity term, respectively, HD is the number of hidden units, ρ is a sparsity parameter (typically a small value close to zero), $\rho_j = (1/N)\sum_{i=1}^{N} y_j\left(x^{(i)}\right)$ is the average activation of hidden unit j over the training set, and $KL(\rho\|\rho_j)$ is the Kullback–Leibler (KL) divergence, which is defined as follows:

$$\mathrm{KL}\left(\rho \| \rho_j\right) = \rho \log \frac{\rho}{\rho_j} + (1-\rho)\log \frac{1-\rho}{1-\rho_j}. \tag{5}$$

The KL divergence fastens the sparsity restraint on the coding procedure. Gradient-based procedures, such as stochastic gradient descent algorithms, can be used to solve this optimization problem.

Stacked autoencoder

A SAE is actually a concatenation of autoencoders which the outputs of the autoencoder stacked on the layer below are wired to the inputs of the successive layer (Bengio et al. 2007). More specifically, for a SAE with L layers, the first layer is trained using the training set as the input. After obtaining the first hidden layer, the output of the kth $(k < L)$ hidden layer is utilized as the input for the $(k + 1)$ hidden layer. Using this method, sequential autoencoders can be stacked hierarchically. Each hidden layer is a higher-level abstraction of the previous layer, and the last hidden layer contains high-level structure and representative information of the input, which are more effective for the successive prediction (Wang et al. 2016).

To employ the SAE model for air quality predictions, a real-value predictor must be added on the top layer. In this paper, a logistic regression (LR) layer was embedded into the network for real-value air quality predictions. The logistic regression model could also be replaced with other regression models, such as the SVR. The SAE model combined with the LR predictor constitutes the entire deep architecture model for air quality predictions, as illustrated in Fig. 2.

Training algorithm

The BP algorithm with the gradient-based optimization technique is widely used for training neural networks (Barnard 1992). Unfortunately, deep networks trained in this manner are known to have poor performance (Hanson and Giles 1993; Kambhatla and Leen 1997; Tenenbaum et al. 2000). Deep networks with large initial weights usually lead to poor local minima, whereas deep networks with small initial weights produce tiny gradients in the bottom layers, which decrease the applicability of training networks with numerous hidden layers (Hinton and Salakhutdinov 2006). To solve this difficulty, Hinton (2006) proposed a greedy layer-wise unsupervised learning technique that can train deep networks effectively. The key idea to use this technique is to pre-train the deep network layer-by-layer in a bottom–up manner. After the pre-training stage, the BP algorithm can be used to fine-tune the entire network's parameters in a top–down fashion. Our training procedure is based on the studies by Hinton et al. (2006) and Bengio et al. (2007), which are provided below.
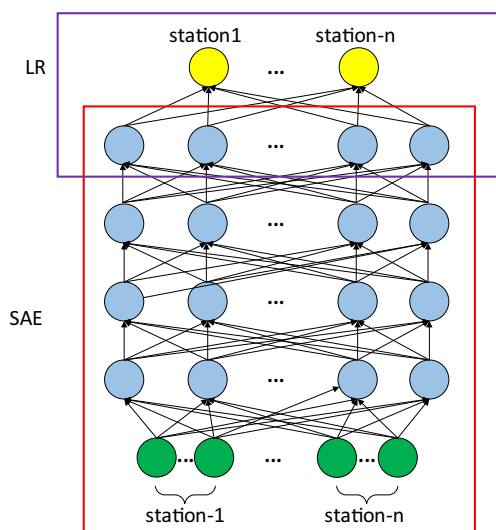
Algorithm 1. Training SAE

For a training sample X and the preset number of hidden layers L and the number of nodes in each hidden layer, initialize the network parameters (i.e., the pre-training epochs, the pre-training learning rate, the fine-tuning epochs, the fine-tuning learning rate, and the mini-batch size).

Step 1 Pre-training the SAE

— Set the weight parameters λ and μ. Randomly initialize the weight matrices and bias vectors.

— Train the first hidden layer using the training set as the input.

— Train the successive hidden layers in a greedy layer-wise manner while using the output of the kth hidden layer as the input for the $(k + 1)$th hidden layer.

Step 2 Fine-tuning the whole network

— Use the output of the last hidden layer as the input for the logistic regression layer.

— Randomly initialize $\{W^{L+1}, b^{L+1}\}$.

— Use the BP algorithm with the gradient-based optimization technique to update the whole network's parameters in a top–down fashion.

Experiment and results

Data description

The hourly PM2.5 concentration data for Beijing City from 2014/1/1 to 2016/5/28 at 12 air quality monitoring stations were downloaded from the Ministry of Environmental Protection of China (http://datacenter.mep.gov.cn/). The PM2.5 concentration of all these stations was measured using a Thermo Fisher 1405F detector based on the tapered element oscillating microbalance (TEOM) method. Figure 3 shows the distribution of these air quality monitoring stations.

Fig. 2 Deep architecture model for air quality predictions: Stacked autoencoders are at the bottom for feature extraction, and a logistic regression layer is at the top for real-value predictions

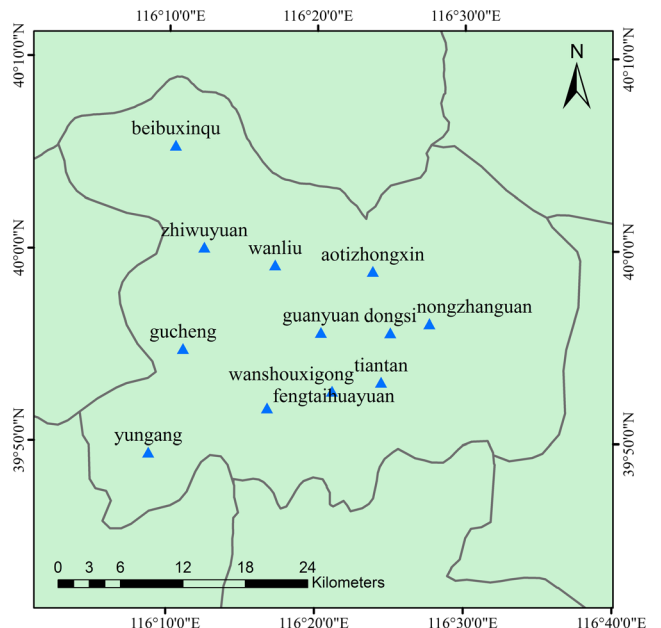**Fig. 3** Distribution of the air quality monitoring stations in Beijing City

Table 1 Average PM2.5 concentration (μg/m³) for each station in different seasons

Station name	Spring	Summer	Autumn	Winter	Average
Dongsi	81.30	68.62	87.84	100.21	84.95
Tiantan	74.67	65.61	85.90	99.06	81.41
Guanyuan	77.51	66.31	89.99	96.33	82.59
Wanshouxigong	77.74	62.63	90.23	106.33	84.63
Aotizhongxin	81.28	66.23	91.87	92.79	83.24
Nongzhanguan	80.42	60.86	94.74	104.35	85.42
Wanliu	77.05	64.78	86.06	98.71	82.09
Beibuxinqu	83.15	58.67	80.49	106.33	83.51
Zhiwuyuan	70.42	59.93	76.15	81.55	72.37
Fengtaihuayuan	85.98	63.27	99.18	119.09	92.47
Yungang	79.55	57.77	88.83	107.27	84.22
Gucheng	81.03	64.09	91.25	100.24	84.52

Table 2 Parameters for our air quality prediction architecture

Parameters	Value set
Time intervals	4, 6, 8,10, 12
Number of layers	1, 2, 3, 4
Number of nodes	100, 200, 300, 400, 500

This dataset contains 20,196 records for each station. Seasonal statistical data are shown in Table 1. In our experiment, we randomly selected 60 % of the data as the training set, 20 % as the validation set, and the remaining 20 % as the test set.

Index of performance

To evaluate the performance of the proposed model, we adopted three performance indexes: the root-mean-square error (RMSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE). These indexes are calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_i - P_i)^2}, \tag{6}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|O_i - P_i|, \tag{7}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|O_i - P_i|}{O_i}. \tag{8}$$

where O_i denotes the observed air quality, P_i denotes the predicted air quality, and N denotes the number of evaluation samples. The RMSE and MAE were used to evaluate the absolute error, while the MAPE was used to measure the relative error. The
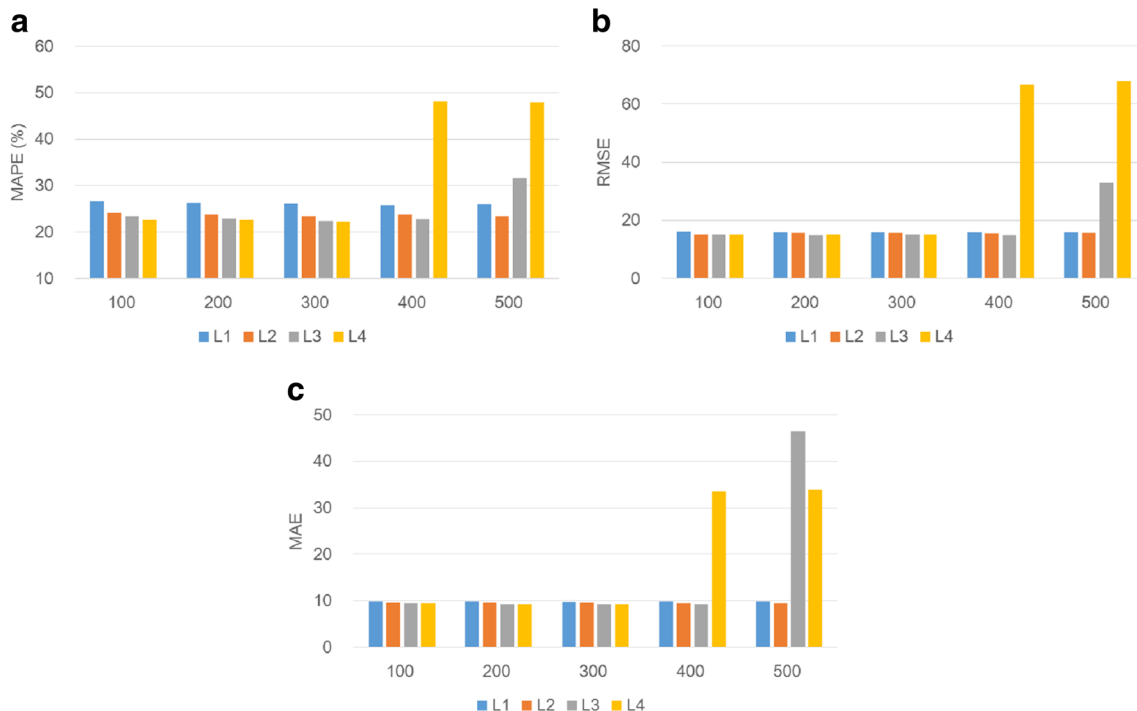

Fig. 4 Performance with various network sizes: (a) MAPE, (b) RMSE, and (c) MAE

Table 3 Effect of the time intervals (layer = 3, nodes = 300)

Time intervals	RMSE	MAE	MAPE (%)
4	15.39	8.60	26.52
6	15.60	8.77	26.54
8	14.96	9.00	21.75
10	17.32	9.50	25.50
12	17.71	9.28	26.62

Table 4 Prediction performance for each station

Station name	RMSE	MAE	MAPE (%)
Dongsi	14.67	9.21	23.97
Tiantan	14.08	8.82	20.65
Guanyuan	14.00	8.44	18.60
Wanshouxigong	15.99	9.24	24.58
Aotizhongxin	14.41	8.85	18.75
Nongzhanguan	16.22	9.93	22.10
Wanliu	13.83	8.54	19.77
Beibuxinqu	16.31	9.88	24.86
Zhiwuyuan	16.00	8.87	27.32
Fengtaihuayuan	16.26	9.76	23.31
Yungang	15.64	9.76	23.42
Gucheng	16.29	9.89	22.21

former reflects the extremum effect and error range of the predicted values, and the latter reflects the specificity of the average predicted value (Chen et al. 2013). The optimal structure of our model was determined when the MAPE was minimized.

Deep architecture structure

Our spatiotemporal deep learning (STDL) model contains several parameters that must be determined to build the architecture, including the size of the input layer, the number of hidden layers, and the number of hidden units in each hidden layer. For the input layer, we used the data collected from all stations as the input; thus, the model could be built upon a monitoring network that considers spatial correlations. Furthermore, with respect to the temporal relationship of the air quality, we used the air quality data at previous time intervals as the inputs (i.e., $t − 1, t − 2, ..., t − r$) to predict the air quality at time interval t. Thus, the proposed model inherently accounts for the spatial and temporal correlations of the air quality data. The dimensions of the input space and output are mr and m, respectively, where m is the number of stations.

We chose the time intervals r from the set {4, 6, 8, 10, 12}; thus, the input dimensions vary from 48 to 144. The numbers of layers were selected from the set {1, 2, 3, 4}. For simplicity, the number of nodes in each hidden layer was set equivalent and selected from the set {100, 200, 300, 400, 500}. All parameters for our prediction model are listed in Table 2. Moreover, the number of training epochs and the learning rate

are also important during the learning phase because the reconstruction error usually increases dramatically with a large learning rate and the model would overfit the training data when the number of epochs is too large. In our experiment, we set the initial pre-training and fine-tuning learning rate to 2 and the scaling learning rate to 0.9995.

As we tested the effect of each parameter, the other parameters were kept fixed. In this step, the validation set was used to evaluate the performance. Better parameter configurations could be identified using a grid search or other heuristic searching methods; however, due to the large search spaces, these methods would be tedious and computationally prohibitive (Huang et al. 2014). Thus, a random search of a fixed set was the preferred method in our experiments.

First, we inspected the effect of various network sizes, which represents one of the most typical problems in neural network design. The training time and the generalization capability of neural network models are highly affected by the network size parameters. In this experiment, the MAPE was the main evaluation index. Figure 4a–c shows the MAPE, RMSE, and MAE values for the various network sizes (i.e.,

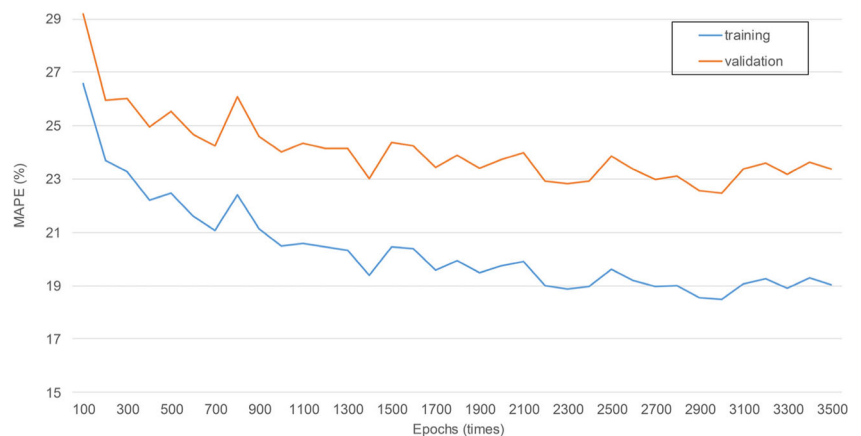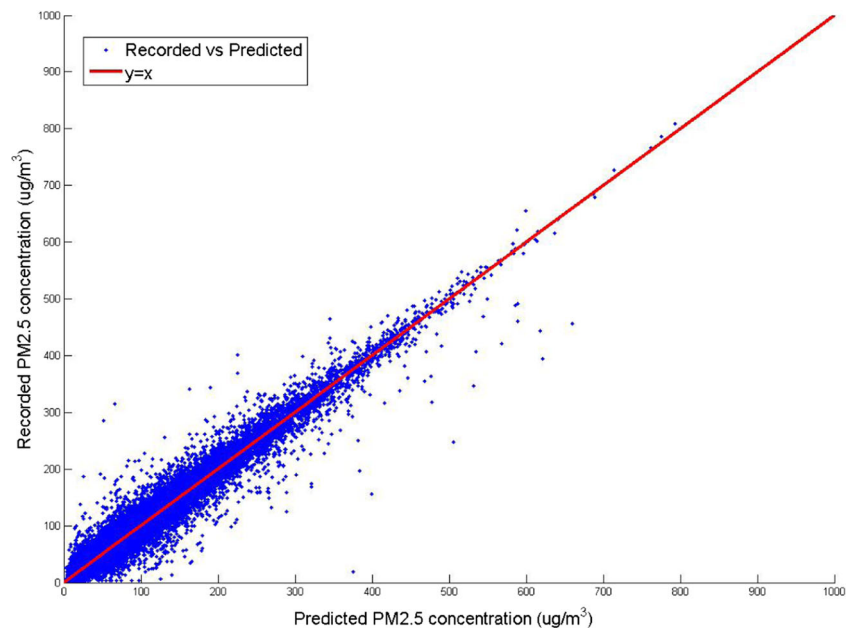**Fig. 5** Effects of the fine-tuned epochs

Fig. 6 Predicted and recorded values of the test set at the Guanyuan station

the number of layers and the number of nodes in each layer). Figure 4a–c shows that the performance could be improved by increasing the number of hidden layers from one to four. High-level air quality features are inherently learned in this manner. However, deeper structures do not have an advantage over a four-layer structure, and models with a structure that is too complex present the issue of overfitting.

Figure 4a–c also shows that increasing the number of nodes in each layer can slightly improve the performance. When the number reached 300, our model presented the best performance. More nodes in each layer would unnecessarily increase the training time and result in overfitting. These phenomena can be easily demonstrated using a four-layer structure with 400 or 500 nodes in each layer because the validation error increases rapidly. To maintain efficiency and accuracy, a structure with three hidden layers and 300 nodes in each layer were used in subsequent experiments.

Next, we tested the effect of different time intervals, as shown in Table 3. A large r would increase the size of the input layer and provide a sufficient number of temporally correlated features, although it increases the training time. The prediction performance obviously increased initially but failed to show improved performance after $r = 8$. If k is greater than 8, then additional latent unrelated inputs make it more

difficult for the complicated architecture to learn a good representation.

Finally, we investigated the effect of fine-tuning epochs. Figure 5 shows the accuracy curve (measured by the MAPE) on the training set and the validation set as a function of the number of epochs. When the epochs were less than 3000, an increase in the number of epochs obviously decreased the training and validation errors. When the epochs were greater than 3000, the model appeared to be overfit, and the generalization capability did not improve but weakly fluctuated. Because a large number of epochs lead to a large temporal cost, we found that the optimal number of fine-tuned epochs at which the training and validation errors converged was 3000 in our experiment.

Results and discussion

First, we evaluated the spatial stability of our STDL model. The predictive performance for each station is shown in Table 4, which indicates that our STDL model showed different predictive performances for these stations. In detail, the RMSE varied from 13.83 to 16.31 μg/m³, the MAE varied from

Table 5 Prediction performance for each season

Season	RMSE	MAE	MAPE (%)
Spring	15.15	9.11	22.95
Summer	15.28	9.28	22.29
Autumn	15.33	9.18	22.42
Winter	15.64	9.53	22.04

Table 6 PM2.5 air quality levels

Rank	Range (μg/m³)	Description
L1	<35	Good
L2	(35, 75)	Moderate
L3	(75, 115)	Unhealthy for sensitive groups
L4	(115, 150)	Unhealthy
L5	(150, 250)	Very unhealthy
L6	>250	Hazardous

Table 7 Predicted and recorded air quality levels (in percent)

Predicted	Recorded						
	L1	L2	L3	L4	L5	L6	Count
L1	30.64	2.61	0.05	0.01	0	0	33.3
L2	2.41	20.26	2.76	0.06	0.03	0	25.52
L3	0.08	1.98	12.14	2.03	0.14	0	16.36
L4	0.01	0.11	1.47	6.81	1.63	0	10.04
L5	0	0.02	0.09	0.87	8.63	0.56	10.17
L6	0.01	0	0	0.01	0.42	4.18	4.61
Count	33.15	24.96	16.51	9.79	10.84	4.75	100

8.44 to 9.33 μg/m^3, and the MAPE varied from 18.60 to 27.32 %. The Guanyuan station (No. 3) had the best performance with the lowest MAPE value of 18.60 %, the lowest MAE value of 8.44 μg/m^3, and a low RMSE value of 14 μg/m^3. The prediction results are shown in Fig. 6, which indicates that the predicted data are generally consistent with the recorded data. The R^2 value between the recorded and predicted hourly PM2.5 concentrations in this testing phase indicated that 98.24 % of the explained variance was captured by the model. The Zhiwuyuan station (No. 9) had the highest relative error and a MAPE value higher than 25 %, which was mainly because this station is located at the border of an urban area and presents only limited air pollutants, such as traffic pollutants. Therefore, this station has relatively good air quality and lower average PM2.5 concentrations (Table 1). Considering that our model produced similar absolute errors (RMAE and MAE) for each station, the MAPE value was higher at the Zhiwuyuan station.

Next, we tested the temporal stability of our model. We calculated the performance index for the four seasons, and the results are shown in Table 5, revealing that our model presented a consistent performance in each season. This feature is beneficial because it indicates that a separate model is not required for each season.

Next, we evaluated the rank prediction performance of our model. According to the National Technical Regulation on the Ambient Air Quality Index (see Table 6), we calculated the recorded and predicted rank rate, which is shown in Table 7. Each row shows the predicted air quality rank ratio, and each column contains the recorded air quality rank ratio. Table 7

shows that the prediction rank rate of our model was high for each air quality rank, and the overall prediction rank accuracy rate was 82.66 %.

Finally, we compared the performance of the proposed STDL model with that of the STANN model, the SVR model, and the ARMA model. These models were trained and tested using the same training and testing sets applied for the STDL model; however, the input data might have been slightly different. The STANN model uses the same inputs as our STDL model, which predicts the air quality of all stations simultaneously based on the spatiotemporal correlations of the input data. The main difference between the STANN model and our STDL model is that the STANN model does not use the greedy layer-wise unsupervised learning algorithm to pretrain the deep network. We conducted the prediction tasks for each station separately for the SVR and ARMA methods, which are merely time series prediction models, using data from a single station as the input. The results are shown in Table 8.

Table 8 reveals that the STDL model presented more accurate air quality predictions than the STANN, SVR, and ARMA models and had lower RMSE, MAE, and MAPE values. Table 8 indicates that the two spatiotemporal models (STDL and STANN) had higher accuracy than the time series models (ARMA and SVR), which shows that spatial correlations are important for air quality predictions. Moreover, a comparison of the performance of the two spatiotemporal models showed that the MAPE of the STDL decreased by 5.12 % compared with that of the STANN, indicating that the deep architecture method with unsupervised pre-training can automatically learn better features than shallow models, thus improving the prediction performance.

Conclusion

In this paper, a spatiotemporal deep learning-based model was developed for air quality prediction. This model consists of a stacked autoencoder model at the bottom for unsupervised feature extraction and a logistic regression model at the top for real-value regression. Compared with existing methods that generally model the shallow structure of air quality data, the proposed method can effectively extract latent air quality feature representations from air quality data, especially nonlinear spatial and temporal correlations. Compared with traditional time series air quality prediction models, our model was able to predict the air quality of all monitoring stations simultaneously, and it showed a satisfactory seasonal stability. We evaluated the performance of the proposed method and compared it with the performance of the STANN, ARMA, and SVR models, and the results showed that the proposed method was effective and outperformed the competitors.

Table 8 Prediction performance for the STDL, STANN, SVR, and ARMA models

Methods	RMSE	MAE	MAPE (%)
STDL	14.96	9.00	21.75
STANN	16.19	10.04	26.87
ARMA	24.40	13.05	27.54
SVR	22.04	11.14	28.45

References

Baklanov A, Mestayer PG, Clappier A, Zilitinkevich S, Joffre S, Mahura A, Nielsen NW (2008) Towards improving the simulation of meteorological fields in urban areas through updated/advanced surface fluxes description. Atmos Chem Phys 8:523–543. doi:10.5194/acp-8-523-2008

Barnard E (1992) Optimization for training neural nets. IEEE Trans Neural Netw 3:232–240. doi:10.1109/72.125864

Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2:1–127

Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks. Adv Neural Inf Proces Syst 19:153

Boureau Y, Cun YL (2008) Sparse feature learning for deep belief networks. In: Platt JC, Koller D, Singer Y, Roweis ST (eds) Advances in Neural Information Processing Systems 20, Proceedings of Neural Information Processing Systems (NIPS 2007)

Box GEP, Jenkins GM (1970) Time series analysis, forecasting, and control. Holden-Day, San Francisco

Bravo MA, Bell ML (2011) Spatial heterogeneity of PM10 and O3 in São Paulo, Brazil, and implications for human health studies. J Air Waste Manage Assoc 61:69–77. doi:10.3155/1047-3289.61.1.69

Bruckman L (1993) Overview of the enhanced geocoded emissions modeling and projection (enhanced GEMAP) system. In: Proceedings of the Air & Waste Management Association's regional photochemical measurements and modeling studies conference, San Diego, CA

Chan TH, Jia K, Gao S, Lu J, Zeng Z, Ma Y (2015) PCANet: a simple deep learning baseline for image classification? IEEE Trans Image Process 24:5017–5032. doi:10.1109/TIP.2015.2475625

Chen Y, Shi R, Shu S, Gao W (2013) Ensemble and enhanced PM10 concentration forecast model based on stepwise regression and wavelet analysis. Atmos Environ 74:346–359. doi:10.1016/j.atmosenv.2013.04.002

Chen J, Lu J, Avise JC, DaMassa JA, Kleeman MJ, Kaduwela AP (2014) Seasonal modeling of PM 2.5 in California's San Joaquin valley. Atmos Environ 92:182–190. doi:10.1016/j.atmosenv.2014.04.030

Cheng T, Wang J, Li X (2007) The support vector machine for nonlinear spatio-temporal regression. Proc Geocomputation 2007

Coats Jr CJ (1996) High-performance algorithms in the sparse matrix operator kernel emissions (SMOKE) modeling system. In: Proc. Ninth AMS Joint Conference on Applications of Air Pollution Meteorology with A&WMA, Amer. Meteor. Soc., Atlanta, GA

Collobert R, Weston J (2008) A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning. ACM, New York, pp 160–167

Díaz-Robles LA, Ortega JC, Fu JS, Reed GD, Chow JC, Watson JG, Moncada-Herrera JA (2008) A hybrid ARIMA and artificial neural networks model to forecast particulate matter in urban areas: the case of Temuco, Chile. Atmos Environ 42:8331–8340. doi:10.1016/j.atmosenv.2008.07.020

Gardner MW, Dorling SR (1998) Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmos Environ 32:2627–2636. doi:10.1016/S1352-2310(97)00447-0

Goyal P, Chan AT, Jaiswal N (2006) Statistical models for the prediction of respirable suspended particulate matter in urban cities. Atmos Environ 40:2068–2077. doi:10.1016/j.atmosenv.2005.11.041

Guocai Z (2004) Progress of weather research and forecast (WRF) model and application in the United States. Meteorol Mon 12:5

Hájek P, Olej V (2012) Ozone prediction on the basis of neural networks, support vector regression and methods with uncertainty. Ecol Inform 12:31–42. doi:10.1016/j.ecoinf.2012.09.001

Hanson SJ, Giles CL (1993) Proceedings, advances in neural information processing systems 5. Morgan Kaufmann Publishers, San Francisco

Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313:504–507. doi:10.1126/science.1127647

Hinton GE, Osindero S, Teh YW, Fast A (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554. doi:10.1162/neco.2006.18.7.1527

Hooyberghs J, Mensink C, Dumont G, Fierens F, Brasseur O (2005) A neural network forecast for daily average PM concentrations in Belgium. Atmos Environ 39:3279–3289. doi:10.1016/j.atmosenv.2005.01.050

Huang W, Song G, Hong H, Xie K (2014) Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans Intell Transp Syst 15:2191–2201. doi:10.1109/TITS.2014.2311123

Jeong JI, Park RJ, Woo J, Han Y, Yi S (2011) Source contributions to carbonaceous aerosol concentrations in Korea. Atmos Environ 45:1116–1125. doi:10.1016/j.atmosenv.2010.11.031

Jerrett M, Burnett RT, Ma R, Pope CA, Krewski D, Newbold KB, Thurston G, Shi Y, Finkelstein N, Calle EE, Thun MJ (2005) Spatial analysis of air pollution and mortality in Los Angeles. Epidemiology 16:727–736. doi:10.1097/01.ede.0000181630.15826.7d

Kambhatla N, Leen TK (1997) Dimension reduction by local principal component analysis. Neural Comput 9:1493–1516. doi:10.1162/neco.1997.9.7.1493

Kim Y, Fu JS, Miller TL (2010) Improving ozone modeling in complex terrain at a fine grid resolution: part I–examination of analysis nudging and all PBL schemes associated with LSMs in meteorological model. Atmos Environ 44:523–532. doi:10.1016/j.atmosenv.2009.10.045

Kracht O, Parravicini F, Gerboles M (2015) Temporal trends of spatial correlation within the PM10 time series of the AirBase ambient air quality database. Int J Environ Pollut 58:63–78

Künzli N, Kaiser R, Medina S, Studnicka M, Chanel O, Filliger P, Herry M, Horak F, Puybonnieux-Texier V, Quénel P, Schneider J, Seethaler R, Vergnaud JC, Sommer H (2000) Public-health impact of outdoor and traffic-related air pollution: a European assessment. Lancet 356:795–801. doi:10.1016/S0140-6736(00)02653-2

Kurt A, Oktay AB (2010) Forecasting air pollutant indicator levels with geographic models 3 days in advance using neural networks. Expert Syst Appl 37:7986–7992. doi:10.1016/j.eswa.2010.05.093

Lal B, Tripathy SS (2012) Prediction of dust concentration in open cast coal mine using artificial neural network. Atmos Pollut Res 3:211–218

Legendre P (1993) Spatial autocorrelation: trouble or new paradigm? Ecol 74:1659–1673. doi:10.2307/1939924

Li C, Hsu NC, Tsay S (2011) A study on the potential applications of satellite data in air quality monitoring and forecasting. Atmos Environ 45:3663–3675. doi:10.1016/j.atmosenv.2011.04.032

Lv Y, Duan Y, Kang W, Li Z, Wang F (2015) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transport Syst: 1–9. doi:10.1109/TITS.2014.2345663

Martin RL, Oeppen JE (1975) The identification of regional forecasting models using space: time correlation functions. Transactions Inst Br Geographers:95–118. doi:10.2307/621623

Mohamed AR, Sainath TN, Dahl G, Ramabhadran B, Hinton GE, Picheny MA (2011) Deep belief networks using discriminative features for phone recognition. In: 2011 I.E. International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Nguyen VA, Starzyk JA, Goh WB, Jachyra D (2012) Neural network structure for spatio-temporal long-term memory. IEEE Trans Neural Netw Learn Syst 23:971–983. doi:10.1109/TNNLS.2012.2191419

Nieto PG, Combarro EF, Del Coz Díaz JJ, Montañés E (2013) A SVM-based regression model to study the air quality at local scale in Oviedo urban area (Northern Spain): a case study. Appl Math Comput 219:8923–8937. doi:10.1016/j.amc.2013.03.018

Poultney C, Chopra S, Cun YL (2006) Efficient learning of sparse representations with an energy-based model. In: Schölkopf B, Platt J, THofmann T (eds) Advances in neural information processing systems 19, proceedings of the 2006 conference. MIT Press, Cambridge, MA, pp. 1137–1144

Prybutok VR, Yi J, Mitchell D (2000) Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations. Eur J Oper Res 122:31–40. doi:10.1016/S0377-2217(99)00069-7

Saide PE, Carmichael GR, Spak SN, Gallardo L, Osses AE, Mena-Carrasco MA, Pagowski M (2011) Forecasting urban PM10 and PM2. 5 pollution episodes in very stable nocturnal conditions and complex terrain using WRF–Chem CO tracer model. Atmos Environ 45:2769–2780. doi:10.1016/j.atmosenv.2011.02.001

Sánchez AB, Ordóñez C, Lasheras FS, de Cos Juez FJ, Roca-Pardiñas J (2013) Forecasting SO2 pollution incidents by means of Elman artificial neural networks and ARIMA models. Abstr Appl Anal 2013:1–6. Hindawi Publishing Corporation. doi:10.1155/2013/238259

Silver D, Huang A, Maddison CJ et al (2016) Mastering the game of Go with deep neural networks and tree search. Nature 529:484–489. doi:10.1038/nature16961

Stern R, Builtjes P, Schaap M, Timmermans R, Vautard R, Hodzic A, Memmesheimer M, Feldmann H, Renner E, Wolke R (2008) A model inter-comparison study focussing on episodes with elevated PM10 concentrations. Atmos Environ 42:4567–4588. doi:10.1016/j.atmosenv.2008.01.068

Suárez Sánchez A, García Nieto PJ, Iglesias-Rodríguez FJ, Vilan Vilan JA (2013) Nonlinear air quality modeling using support vector machines in Gijón urban area (northern Spain) at local scale. Int J Nonlin Sci Numer Simul 14:291–305

Tenenbaum JB, De Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290: 2319–2323. doi:10.1126/science.290.5500.2319

Vautard R, Builtjes PHJ, Thunis P, Cuvelier C, Bedogni M, Bessagnet B, Honoré C, Moussiopoulos N, Pirovano G, Schaap M (2007) Evaluation and intercomparison of Ozone and PM10 simulations by several chemistry transport models over four European cities within the CityDelta project. Atmos Environ 41:173–188. doi:10.1016/j.atmosenv.2006.07.039

Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning

Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408

Wang Q, Lin J, Yuan Y (2016) Salient band selection for hyperspectral image classification via manifold ranking. IEEE Trans Neural Netw Learn Syst 27:279–1289

Zhang CY, Chen CLP, Gan M et al (2015) Predictive deep Boltzmann machine for multiperiod wind speed forecasting. IEEE Trans Sustain Energy 6:1416–1425

Zheng Y, Liu F, Hsieh H (2013) U-Air: when urban air quality inference meets big data. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, pp 1436–1444