Article

Multi-Site Multi-Pollutant Air Quality Data Modeling

Min Hu , Bin Liu * , Guosheng Yin

Posted Date: 17 November 2023

doi: 10.20944/preprints202311.1091.v1

Keywords: Air quality; Multi-pollutant prediction; Graph convolutional neural network; Long short term memory

Article

Multi-Site Multi-Pollutant Air Quality Data Modeling

Min Hu [1], Bin Liu [2,*] and Guosheng Yin [3]

[1] School of Finance, Southwestern University of Finance and Economics, Chengdu, China; min_hu@smail.swufe.edu.cn
[2] Center of Statistical Research, School of Statistics, Southwestern University of Finance and Economics, Chengdu, China; liubin@swufe.edu.cn
[3] Department of Statistics and Actuarial Science, The University of Hong Kong, Hong Kong, China; gyin@hku.hk
* Correspondence: liubin@swufe.edu.cn

Abstract: Air quality is one of the most concerning problems in major industrialized cities in the world. Prediction of future air quality is highly relevant to public health. In some big cities, multiple air quality measurement stations are deployed at different locations to monitor air pollutants, such as NO_2, CO, PM 2.5 and PM 10, over time. At every monitoring time stamp t, we observe one $station \times feature$ matrix $\mathbf{x}_t$ of the pollutant data, which represents a spatio–temporal process. Traditional methods on prediction of air quality typically use data from one station or can only predict a single pollutant (such as PM 2.5) at a time, which ignores the spatial correlation among different stations. Moreover, the air pollution data are typically highly nonstationary, We propose a de-trending graph convolutional LSTM (long short term memory) to continuously predict the whole $station \times feature$ matrix in the next 1 to 48 hours, which not only captures the spatial dependency among multiple stations by replacing an inner product with convolution, but also incorporates the de-trending signals (transform a nonstationary process to a stationary one by differencing the data) into our model. Experiments on the air quality data of the city Chengdu and multiple major cities in China demonstrate the feasibility of our method and show promising results.

Keywords: air quality; multi-pollutant prediction; graph convolutional neural network; long short term memory

1. Introduction

The issue of air quality is not just a health concern but also a pressing issue of sustainable development. With the rapid development of the global economy, the challenge of air pollution has become increasingly apparent. The issues of air quality are complicated and have received increasing attention in China. In most cities, multiple monitoring stations are distributed at different locations to report real-time air quality indices. Typically, the levels of air pollutants are periodically recorded by multiple stations (for example, the data used in this paper are recorded every hour). It means that at every time stamp one air quality matrix with the shape of $station \times feature$ can be collected by all the stations, where features include NO_2, CO, PM 2.5, PM 10 etc. Based on this air quality data matrix, an air quality index (AQI) can be calculated to inform public the air quality at present. However, the general public are more interested in prediction of future air quality rather than the real-time reporting. Not only such prediction can benefit people's daily life activities (for example, making a travel plan or avoiding routes of poor air quality) and improve their health by wearing masks to reduce exposure to air pollution, but it also provides policy implications for the government. There are a large number of works for air quality prediction in the literature [1–3]. Most of them are based on the temporal dependency between future states and historical data, such as time series models [2,4,5] and deep neural networks [1,6–8]. However, there are several limitations to the existing methods. First, many models [2,6,7] take the air quality prediction as a single-pollutant regression problem, for example, focusing on only the particulate matter PM 2.5. To predict the level of another pollutant, e.g., the carbon monoxide CO, a different model needs to be trained. Second, to improve the performance for

doi:10.20944/preprints202311.1091.v1

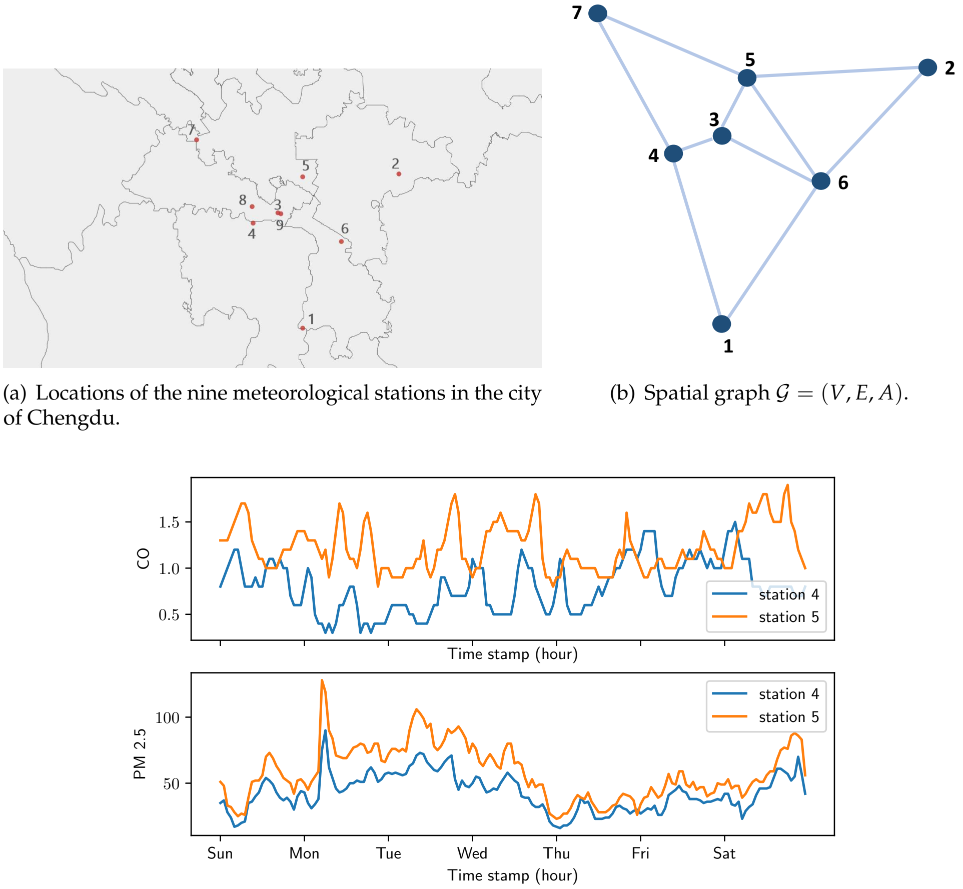
prediction, some methods [5,9] choose to incorporate extra knowledge, such as the weather forecasting results [9] or the traffic data [5]. In practice, it is not convenient to collect the extra information and synchronize them with the air quality data. When data from multiple stations are available, the geographical correlations among these stations are expected to be useful for air quality forecasting [5,9]. However, most existing methods can only deal with the data from one station at a time such that the spatial correlations among multiple stations are ignored [1,2,7] or partially considered [5,9]. Last but not least, the air quality data usually represent high nonstationarity as shown in Figure 2, with the mean of data changing over time, which makes the modeling problem more difficult. Therefore, learning the underlying spatio–temporal features from a nonstationary process is particularly crucial for prediction. We propose to solve the aforementioned problems using a nonstationary diffusion graph convolutional LSTM (long short term memory). In detail, the spatio–temporal characteristics of air quality data from multi-sites motivate us to use the diffusion convolutional LSTM network [10]. The diffusion convolution [11] captures the spatial dependency using bidirectional random walks on the meteorological monitoring sites graph $\mathcal{G} = (V, E, A)$ as shown in Figure 1 (b). In addition, we add a de-trending step to the diffusion convolutional LSTM to accommodate the nonstationarity of the data. We name the proposed model as the long-short de-trending graph convolutional network (LS-deGCN). In time series analysis, the de-trending step is usually implemented by a differencing procedure $\mathbf{x}_t - \mathbf{x}_{t-1}$ [12,13]. As a result, the input of the proposed model involves both $\mathbf{x}_t$ and $\mathbf{x}_t - \mathbf{x}_{t-1}$.

Motivated by the vast applications of LSTM in the area of natural language processing (NLP), we propose two variants of the LS-deGCN, as shown in Figure 3. At every time stamp, there is one $station \times feature$ matrix of data $\mathbf{x} \in \mathbb{R}^{M\times N}$ observed, where M is the number of stations and N is the number of features. Therefore, the final dataset is a 3-dimensional tensor $\mathcal{X} \in \mathbb{R}^{M\times N\times T}$ by stacking all $\mathbf{x}$'s along time, where T is the number of time stamps. Given a window length, for example 3, which is a tuning parameter in our model, we can slice the samples along the third dimension T. For ease of exposition, we omit the first two dimensions and denote $\mathcal{X}[0 : 3] = \mathcal{X}[:, :, 0 : 3]$. The sliced samples are in the form of $\mathbf{x}_1 = \mathcal{X}[0 : 3], \mathbf{x}_2 = \mathcal{X}[1 : 4]$, etc. We propose two different ways of defining the target $\mathbf{y}$, which correspond to the two variants of our models. One is a sequence-to-frame model; that is, $\mathbf{y}_1 = \mathcal{X}[4], \mathbf{y}_2 = \mathcal{X}[5]$, etc; and the other is a sequence-to-sequence model; that is, $\mathbf{y}_1 = \mathcal{X}[1 : 4], \mathbf{y}_1 = \mathcal{X}[2 : 5]$, etc.

The contributions of this paper are threefold:

1. We modify the traditional LSTM by adding a de-trending operation for nonstationary data;
2. We propose to use the diffusion graph convolution to extract the spatial correlations in the air quality data from multi-sites;
3. We propose two different models based on the LS-deGCN to predict air quality at multi-sites and evaluate them on the air quality data in the city of Chengdu and the other data from seven major cities.

The rest of the paper is structured as follows. Section 2 presents a brief review of related works. In Section 3, we introduce the proposed LS-deGCN and its two variants:the sequence-to-frame model and the sequence-to-sequence model. The experimental results are shown in Section 4. Section 5 concludes this paper with some remarks.



(a) Locations of the nine meteorological stations in the city of Chengdu.

(b) Spatial graph $\mathcal{G} = (V, E, A)$.

(c) The records of levels of CO and PM 2.5 from station 4 and station 5 over one week.

Figure 1. Nine stations are spatially distributed in the city of Chengdu and levels of CO and PM 2.5 from station 4 and station 5 are recorded by hours. From panel (a), we observe that station 4 and station 5 are separated by a long geographical distance, while the readings of CO and PM 2.5 by these two stations in panel (c) are strongly correlated with each other. Spatial correlations: the readings from station 3 and station 9 are the same, and thus station 9 is removed. We also delete the data from station 8 because 40% data are missing. Panel (b) illustrates the spatial dependency graph among the seven stations we studied.

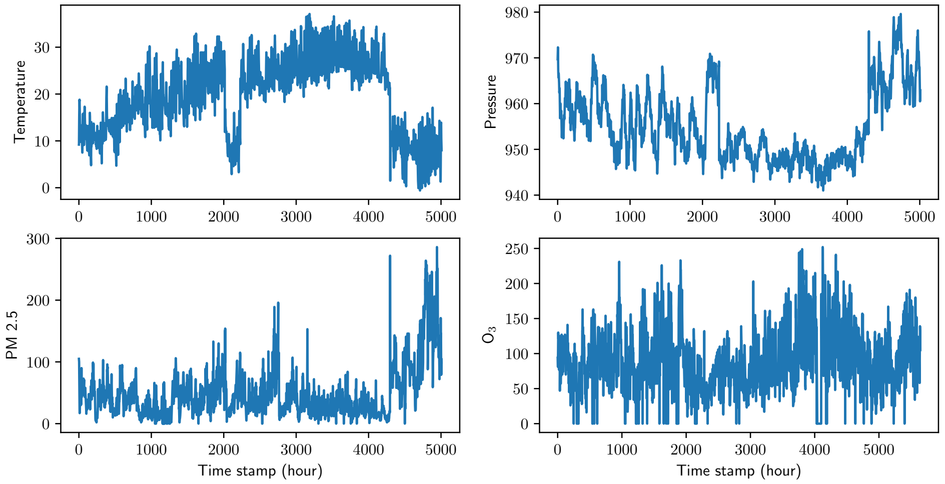
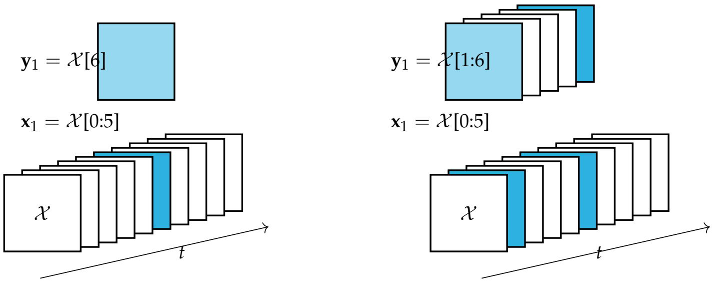


Figure 2. A sketch of nonstationary records of the features of temperature, pressure, PM 2.5, and O_3.


$\mathbf{y}_1 = \mathcal{X}[6]$
$\mathbf{x}_1 = \mathcal{X}[0:5]$
$\mathcal{X}$
t
$\mathbf{y}_1 = \mathcal{X}[1:6]$
$\mathbf{x}_1 = \mathcal{X}[0:5]$
$\mathcal{X}$
t

(a) The seq2frame model

(b) The seq2seq model

Figure 3. Graphical illustration of the two data generation schemes.

2. Literature Review

Traditionally, meteorologists often make air quality prediction based on their empirical knowledge of meteorology. With the development of statistics and machine learning, data-driven methods for air quality prediction are becoming increasingly popular nowadays, which can be typically divided into statistical approaches [2,14–17] and deep learning approaches [6,7,9,10,18,19]. Some existing deep learning models treat the air quality forecasting as a single pollutant regression problem, and thus they only predict one pollutant at a time [2,6,20]. As a result, separate models need to be trained for different pollutants if all of the pollutants are of interest, where each model only focuses on one pollutant. Zhang et al. [2] conducted a statistical analysis of the PM 2.5 data in years 2013–2016 from the city Beijing based on a flexible nonstationary hierarchical Bayesian model. Mukhopadhyay and Sahu [17] proposed a Bayesian spatio–temporal model to estimate the long-term exposure to air pollution levels in England. Since the air quality records are typically monitored over time, Ghaemi et al. [15] designed an LaSVM-based online algorithm to deal with the streaming of the air quality data. Along another direction, the Granger causality has been proposed to analyze the correlations among the air pollution sequences from different monitoring stations [4,5,9]. Suppose that the sequence of air pollutant records from one station is denoted by $\mathbf{x}_t$, and the sequence of a factor (such as the geographical correlation) from another station by $\mathbf{y}_t$, then the mathematical representation of the Granger causality is given by

$$\mathbf{x}_t = \sum_{k=1}^{K} a^k \mathbf{x}_{t-k} + \sum_{k=1}^{K} b^k \mathbf{y}_{t-k} + \epsilon_t, \tag{1}$$

where a^k is a weight indicating how the width of time window k affects the future evolution, b^k represents the correspondent weight for $\mathbf{x}_t$ and $\mathbf{y}_t$, and ϵ_t is a residual for time series $\mathbf{x}_t$. If $a^k \neq 0$ and $b^k = 0$, it means that the sequence $\mathbf{x}_t$ is caused by its own history. Wang and Song [9] combined the Granger causality with deep learning models and there are also some connections between the Granger causality and LSTM [21].

The deep learning, more specifically, the recurrent neural network (RNN) and LSTM [21] have achieved vast success in the area of NLP [22]. Due to their capability in modeling sequence data, the RNN and LSTM demonstrate good performance in air quality prediction as well [1,8,23]. Guo et al. [23] proposed a multi-variable LSTM based on both temporal and variable level attention mechanisms, which was used to predict the PM 2.5 level in Beijing. Fan et al. [8] also used the LSTM as a framework to predict air quality in Beijing based on the air pollution and meteorological information. In contrast to [23], the data used in [8] are collected from multiple stations, while data from different stations are analyzed separately by ignoring the spatial correlation. In big cities, there are usually more than one stations deployed to monitor the air pollutants and meteorological information. The correlations among readings from different stations are highly informative in forecasting future air quality and thus the spatial information should be incorporated. Xu et al. [24] proposed a multi-scale 3-dimensional tensor decomposition algorithm to handle the spatio–temporal correlation in climate modeling. In deep learning, the convolutional neural network (CNN) has advantages in extracting spatial features, while the RNN has superior performance in processing sequence data. Therefore, it is expected to achieve more accurate prediction by combining convolution and RNN in the analysis of spatial and temporal data. Huang and Kuo [6] proposed to stack a CNN over LSTM to predict the level of PM 2.5. However, this modification leads to a non-time-series model, which may cause power loss in quantifying the sequential air quality data. Wilson et al. [25] proposed a deep learning approach based on the graphical CNN for weather forecasting, where a weighted graph, aiming to capture the spatial correlation, is calculated based on the original features.

In this paper, we propose to model the air quality data from multiple stations with a LS-deGCN, which replaces the fully connected layer in the classical LSTM with graph diffusion convolution. The LS-deGCN architecture can be well adapted to the spatio–temporal data, which was first proposed to analyze the 2-dimensional radar echo map [10]. In addition, our model can accommodate the nonstationarity of the air quality data. Compared with the work of Wilson et al. [25], our method is easier to be implemented and trained. A similar diffusion convolutional recursive neural network was proposed in [26] to model the traffic flow.

3. Proposed Models

3.1. Problem and Research Gap

The LSTM network is generally designed to deal with sequential data [21], which has achieved great success in NLP [22] and video analysis [27]. The LSTM can capture both the long and short contextual dependency of the data via different types of gates. In the classical LSTM, the well-designed gates, for example, the input gate and the forget gate, make the network very powerful to model the temporal correlations of the sequential data.

For the problem of air quality prediction, the data are hourly recorded by multiple stations. At each time stamp, the observed data can be represented by a $station \times feature$ matrix $\mathbf{x} \in \mathbb{R}^{M \times N}$, where M is the number of stations and N is the number of features. The features refer to air pollutants (e.g., CO_2, PM 2.5) and meteorological parameters (e.g., air pressure, air temperature, and air humidity). As shown in Figure 1, the air quality data of the city Chengdu involve 9 monitoring stations and 9 features. The entire data thus can be treated as a 3-dimensional tensor $\mathcal{X} \in \mathbb{R}^{M \times N \times T}$, with respect to the three axes $station \times feature \times time$. The third dimension T corresponds to the number of time stamps, which indicates the sequential nature of the data $\mathcal{X}$.

The air quality data in Chengdu have a typical spatio–temporal pattern, as the data from different stations show strong spatial correlations. Figure 1 (a) are the locations of different monitoring stations. According to the pairwise geographic distances between different stations, we compute the undirected graph $\mathcal{G} = (V, E, A)$ with thresholded Gaussian kernel [28], $A_{ij} = \exp(-\frac{\|v_i - v_j\|}{\sigma^2})$, as shown in Figure 1 (b), where A_{ij} is an element of the adjancent matrix A, v_i and v_j are the i, j-th nodes of $\mathcal{G}$. Figure 1 (C) exhibits the one-week readings of two types of pollutants, i.e., CO and PM 2.5 from station 4 and station 5 respectively. There are clear shifts from station 4 to station 5 in the records of CO and PM 2.5, where the patterns are similar. The area where station 5 is located appears to be more polluted than the area of station 4, and such information is useful for government policy making in different districts. The similar oscillating patterns in the records of the two stations imply that the rows and columns in the matrix $station \times feature$ are correlated and further series correlation over time can also be observed. However, there seems no obvious daily periodicity or “weekends/holidays” effect from the data. Although the classical LSTM possesses powerful capability in modeling time series data, it may not be suitable to capture such type of spatial correlation in the data [8].

3.2. Nonstationary Diffusion Convolutional LSTM

We propose to model the nonstationary $station \times feature \times time$ data with the de-trending diffusion convolutional LSTM (LS-deGCN), which aims to incorporate both the spatio–temporal correlations [10] and nonstationarity into our model. In contrast to the classical LSTM where all the gates are implemented by fully connected neural networks, the proposed model replaces them with diffusion operations [11]. In addition, the differences of the input signals $\mathbf{x}_t - \mathbf{x}_{t-1}$ and corresponding hidden signals $\mathbf{h}_{t-1} - \mathbf{h}_{t-2}$ are input for de-trending as follows,

$$f_t = \sigma\left(W_f * [\mathbf{x}_t, \mathbf{h}_{t-1}] + W_{df} * [\mathbf{x}_t - \mathbf{x}_{t-1}, \mathbf{h}_{t-1} - \mathbf{h}_{t-2}] + W_{cf} \circ \mathbf{c}_{t-1} + b_f\right) \tag{2}$$

$$i_t = \sigma\left(W_i * [\mathbf{x}_t, \mathbf{h}_{t-1}] + W_{di} * [\mathbf{x}_t - \mathbf{x}_{t-1}, \mathbf{h}_{t-1} - \mathbf{h}_{t-2}] + W_{ci} \circ \mathbf{c}_{t-1} + b_i\right) \tag{3}$$

$$\mathbf{c}_t = f_t \circ \mathbf{c}_{t-1} + i_t \circ \tanh\left(W_c * [\mathbf{x}_t, \mathbf{h}_{t-1}] + W_{dc} * [\mathbf{x}_t - \mathbf{x}_{t-1}, \mathbf{h}_{t-1} - \mathbf{h}_{t-2}] + b_c\right) \tag{4}$$

$$\mathbf{o}_t = \sigma\left(W_o * [\mathbf{x}_t, \mathbf{h}_{t-1}] + W_{do} * [\mathbf{x}_t - \mathbf{x}_{t-1}, \mathbf{h}_{t-1} - \mathbf{h}_{t-2}] + W_{co} \circ \mathbf{c}_t + b_o\right) \tag{5}$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh\left(\mathbf{c}_t\right), \tag{6}$$

where $*$ and $\circ$ represent the diffusion convolution and Hadamard product respectively, $[\mathbf{a}, \mathbf{b}]$ is concatenation of vectors **a** and **b**, $\sigma(\cdot)$ is the *sigmoid* function, and $tanh(\cdot)$ is the hyperbolic tangent function. Eqs (2), (3) and (5) correspond to the implementations of the forget gate, input gate, and output gate. Eq (4) is the updating mechanism of the cell state $\mathbf{c}_t$, and the hidden state $\mathbf{h}_t$ is updated by Eq (6). Note that $\{W_f, W_{df}, W_{cf}, b_f\}$ are the weights (corresponding to the feature, de-trending term and hidden state) and bias of the forget gate, and $\{W_i, W_{di}, W_{ci}, b_i\}$ and $\{W_o, W_{do}, W_{co}, b_o\}$ are the counterparts of the input gate and the output gate respectively. Furthermore, $\{W_c, W_{dc}, b_c\}$ are the weights (corresponding to the feature and de-trending term) and bias of the hidden state.

The diffusion convolution [11] is defined as follows,

$$\hat{\mathbf{x}} = \sum_{m=1}^{M} \sum_{k=0}^{K-1} (w_{k1}(D_O^{-1}A)^k + w_{k2}(D_I^{-1}A)^k)\mathbf{x} \tag{7}$$

where $\mathbf{x}$ is an input matrix and $\hat{\mathbf{x}}$ is the corresponding output. A is an adjacent matrix of the M stations as shown in Figure 1 (b), and D_O and D_I are the out-degree and in-degree diagonal matrix of A.

The frame differencing term $\mathbf{x}_t - \mathbf{x}_{t-1}$ is governed by the corresponding coefficient matrices. In Eq (5), we observe that not only does the current output $\mathbf{o}_t$ depend on the current input $\mathbf{x}_t$, the current cell state $\mathbf{c}_t$, and the previous hidden state $\mathbf{h}_{t-1}$, but it also requires the input of the differencing signal $\mathbf{x}_t - \mathbf{x}_{t-1}$. We also concatenate the hidden differencing signal $\mathbf{h}_{t-1} - \mathbf{h}_{t-2}$ with the input differencing $\mathbf{x}_t - \mathbf{x}_{t-1}$.

The main differences between the classical LSTM and our LS-deGCN lie in the diffusion graph convolutional operations and the extra de-trending items $\mathbf{x}_t - \mathbf{x}_{t-1}$ and $\mathbf{h}_{t-1} - \mathbf{h}_{t-2}$. If the graph convolution in Eqs (2)–(6) are substituted with inner products and set $\mathbf{x}_t - \mathbf{x}_{t-1} = 0, \mathbf{h}_{t-1} - \mathbf{h}_{t-2} = 0$, then the LS-deGCN reduces to the classical LSTM. The great success achieved by the graph convolutional neural network is largely owed to the graph convolution which can effectively capture the spatial information of data. This is also the rationale underlying our LS-deGCN to characterize the spatial correlations.

As discussed earlier, the air quality data are typically collected from multiple stations over time, which represent both the spatial and temporal characteristics. The spatial correlations among different stations depend on many complicated factors such as geographical distances, directions and meteorological features, which are very important for air quality prediction [4,5,9]. In the existing works, the spatial correlations are typically inferred manually by resorting to some complicated statistical processes. In this paper, we propose to model the spatial correlations with diffusion graph convolution. In other words, the spatial correlations among different stations can be extracted by the LS-deGCN, which is the most notable difference between our method and the existing models for air quality prediction. This distinctive feature makes our model coherent in the deep learning framework and more powerful for air quality forecasting.

3.3. Two new models

According to the scheme of generating samples and targets as shown in Figure 3, we develop two variants of our modeling framework. To help further elaboration, we first introduce some necessary notation. We first select Δ_t, the width of a slicing window or the length of the samples, which is an analogy to the sentence length in NLP. The value of Δ_t represents how many historical sample observations are used for prediction. Let l denote the time lag from a sample to the target; that is, our goal is to make a prediction of air quality for l hours later. We then propose the following two modeling frameworks.

In the sequence-to-frame (seq2frame) model, we take samples to be slices of the 3-dimensional tensor $\mathcal{X} \in \mathbb{R}^{M \times N \times T}$ along the dimension T. The sample length is Δ_t, the width of the slicing window. The target is taken to be the single frame after the corresponding samples. Since the time lag from a sample to its target is l, the training and testing samples can be generated as follows,

$$\begin{aligned} \mathbf{x}_i &= \mathcal{X}[i-1 : \Delta_t + i - 1], \\ \mathbf{y}_i &= \mathcal{X}[\Delta_t + i + l - 1]. \end{aligned}$$

Figure 3 (a) illustrates the way to generate $\mathbf{x}_i$ and $\mathbf{y}_i$ in the seq2frame model, with $\Delta_t = 5$ and time lag $l = 1$. As a result, we can obtain the first sample and its target, i.e., $\mathbf{x}_1 = \mathcal{X}[0 : 5]$, and $\mathbf{y}_1 = \mathcal{X}[6]$.

The sequence-to-sequence (seq2seq) model is motivated by the seq2seq model in NLP [22], under which the samples and targets are generated as in Figure 3 (b) with both the sample and target of length Δ_t slices. We use the same way of slicing samples $\mathbf{x}$ as that for the seq2frame scheme and define a time l-shifted sequence after $\mathbf{x}$ to be the target $\mathbf{y}$. This data generating scheme leads to

$$\begin{aligned} \mathbf{x}_i &= \mathcal{X}[i-1 : \Delta_t + i - 1], \\ \mathbf{y}_i &= \mathcal{X}[i + l - 1 : \Delta_t + i + l - 1]. \end{aligned}$$

As illustrated in Figure 3 (b), with $\Delta_t = 5$ and time lag $l = 1$, the first sample is $\mathbf{x}_1 = \mathcal{X}[0 : 5]$ and its target is $\mathbf{y}_1 = \mathcal{X}[1 : 6]$.

3.4. Selection of tuning parameters l and Δ_t

The time lag l and window width Δ_t are critical tuning parameters for the proposed models. If the value of l is too small, the overlapping of the information would be too much. In contrast, a larger value of time lag l would make the prediction more difficult. Theoretically, the proposed model can predict the future air quality for any length of the window by setting l to be large enough. In practice, we explore the value of l from 1 hour to 48 hours, which means that we aim to predict the air quality till the day after tomorrow. Intuitively, the more past data are used, the better performance would be expected. In other words, the performance is usually monotonically increasing with the value of Δ_t.

4. Experiments and Results

4.1. Baselines

We choose the following three models as the baselines for comparisons.

1. **Linear regression**: This is one of the most commonly used approaches to modeling the relationship between a dependent variable y and covariates $\mathbf{x}$.
2. **Support vector regression**: Equipped with a radial basis kernel, it extends linear regression by controlling how much error in regression is acceptable.
3. **LSTM sequence-to-scalar** (seq2scalar): Samples under this model are constructed in the same way as those under the seq2seq model. The difference is that we take the target y as one of the nine pollutants one by one; that is, we need to train nine separate models for the nine pollutants.

The three baseline methods are all one-pollutant regression procedures; that is, we need to train separate models on each individual pollutant at a time. Taking the linear regression as an example, to compare with the proposed model that predicts the whole $station \times feature$ map, we need to separately train nine linear regression models corresponding to the six air pollutants (NO_2, CO, SO_2, O_3, PM 2.5, and PM 10) and the three meteorological measurements.

We use the root mean squared error (RMSE), accuracy, and mean absolute error (MAE) as assessment criteria for prediction,

$$\begin{aligned}
\text{RMSE} &= \sqrt{\frac{\sum_{i=1}^{n}(\hat{\mathbf{y}}_i - \mathbf{y}_i)^2}{n}}, \\
\text{Accuracy} &= 1 - \frac{\sum_{i=1}^{n}|\hat{\mathbf{y}}_i - \mathbf{y}_i|}{\sum_{i=1}^{n}\mathbf{y}_i}, \\
\text{MAE} &= \frac{\sum_{i=1}^{n}|\hat{\mathbf{y}}_i - \mathbf{y}_i|}{n},
\end{aligned}$$

where $\mathbf{y}_i$ is the ground truth, $\hat{\mathbf{y}}_i$ is the predicted value, and n is the size of the testing dataset.

4.2. Data description and preprocessing

The Chengdu dataset is composed of the air quality records from January 1, 2013 to December 31, 2016 from nine monitoring stations in the city of Chengdu. Because stations 3 and 9 were very close to each other and their records were exactly the same, we removed the redundancy by dropping the data from station 9. Moreover, we deleted the data from station 8 because over 40% readings for CO, SO_2 and O_3 were missing which mainly occurred between June 2014 to January 2016 due to the sensor dysfunction. As a result, we only used data from seven stations, and there are 35,064 instances for each station. Each air quality instance consists of the concentration of six air pollutants: NO_2, CO, SO_2, O_3, PM 2.5, PM 10, and three meteorological measurements including air pressure, air temperature, and air humidity. Therefore, the observed data are in the form of $\mathcal{X} \in \mathbb{R}^{M \times N \times T}$, where $M = 7$, $N = 9$, and $T = 35,064$. We use the linear interpolation to fill in the missing values of $\mathcal{X}$ on the time domain for each column. For example, we can interpolate the missing $\mathbf{x}_i$ with the values of $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$. After interpolation, we normalize all the data into the range of $[0,1]$ using the minmax normalization.

The training and testing samples for baselines are generated in the same way as those for the seq2frame model. The difference is how to extract the target. In the seq2frame architecture, the network receives one frame (matrix) as a target, while the targets of the baseline models are all scalars. Instead of using the whole frame as the target, we take a statistic (e.g., the mean) of the measurement of one pollutant within that frame as the target. Taking the linear regression as an example, we model the air quality prediction as a one-pollutant linear regression problem. Suppose that we fit a linear regression model to predict the PM 2.5 emission, then we can use the mean or the median of the PM 2.5 level within that frame as the target.

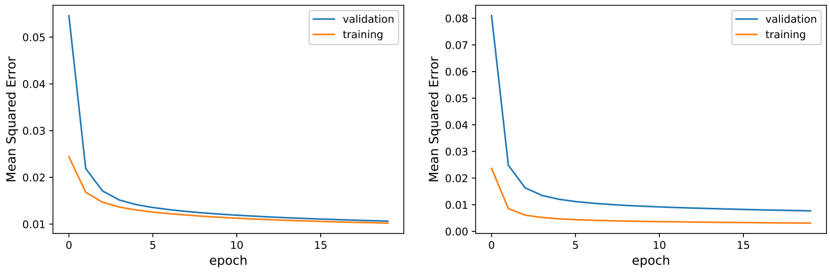
4.3. Training

After data normalization, we extracted the training samples and testing samples from the dataset $\mathcal{X}$. We set the window width to be $\Delta_t = \{24, 48\}$ hours to strike a balance between the length of contextual information and the LSTM model complexity. As a result, every sample is a consecutive record of air quality measurements for one day or two days. We partitioned the data into three sets as shown in Table 1. We constructed a two-layer LS-deGCN by stacking two one-layer LS-deGCNs.

Table 1. Partition for training, validation, and testing datasets.

Datasets	Description
Training	Data from January 1, 2013 to December 31, 2015
Validation	Data from January 1, 2016 to June 1, 2016
Testing	The remaining data

Figure 4 (a) and (b) report the training and validation errors under the seq2frame and seq2seq models respectively. Both models converged in 10 epochs, while the patterns of their convergence are different. Under the seq2frame model, both the training error and validation error drop sharply within the first three epochs and then the gap between them gradually decreases. By contrast, the gap between the training error and validation error under the seq2seq model remains at a stable level even after 15 epochs. The different error patterns may be due to the different ways of extracting the target variable **y** under the two models, as shown in Figure 3.



(a) The seq2frame model

(b) The seq2seq model

Figure 4. Mean squared errors for the training and validation datasets under (a) the seq2frame model, and (b) the seq2seq model, with the time lag $l = 48$ and the window width $\Delta_t = 48$.

4.4. *Visualization of predictions*

To visualize part of the predicted results, Figure 5 shows four $station \times feature$ air quality maps randomly selected from the testing dataset, as well as the corresponding predicted $station \times feature$ frames by the seq2seq model with $l = 24$ and $\Delta_t = 48$. For each square of the $station \times feature$ matrix, the dark blue color represents a larger value and the light blue color indicates a smaller value.

The far right-end three columns (corresponding to the three meteorological measurements: air pressure, air temperature, and air humidity) of the ground truth map are almost identical across seven stations. In contrast, the levels of air pollutants recorded by different stations are quite different. For example, the levels of O_3 (the third column) reported at different locations show very different patterns, and in the third frame station 2 and 5 reported a much higher value than other stations. The proposed model can make a accurate prediction of O_3 for each station. We also observe that the values of air pressure in the third frame are much lower than the other three frames, and the level of suspended particulate matter, PM 10 and PM 2.5, in the last frame are serious, with the level of PM 10 from station 1 and 6 being the top 2 highest. These trends are all correctly predicted by our proposed seq2seq model. From the visualization results, we conclude that (1) the patterns of the nine air quality measurements are different from one another; and (2) the proposed model can make an accurate prediction based on all of the nine measurements from seven stations.

To examine the proposed air quality model across different cities, we conducted another experiment with seven stations from seven major cities in China, including Beijing, Shanghai, Chengdu, Wuhan, Guangzhou, Xi'an, and Nanjing. The data were collected daily (instead of hourly) on seven pollutants (AQI, PM 2.5, PM 10, SO_2, CO, NO_2, and O_3) from December 2, 2013 to February 29, 2020 for each city. Figure 6 shows that the performance of air quality prediction for seven cities is not as good as that for seven stations from the same city of Chengdu. One possible reason is that the spatial correlation among seven stations in Chengdu is much stronger than the seven cities that are far away from each other, and thus the proposed LS-deGCN is more effective.

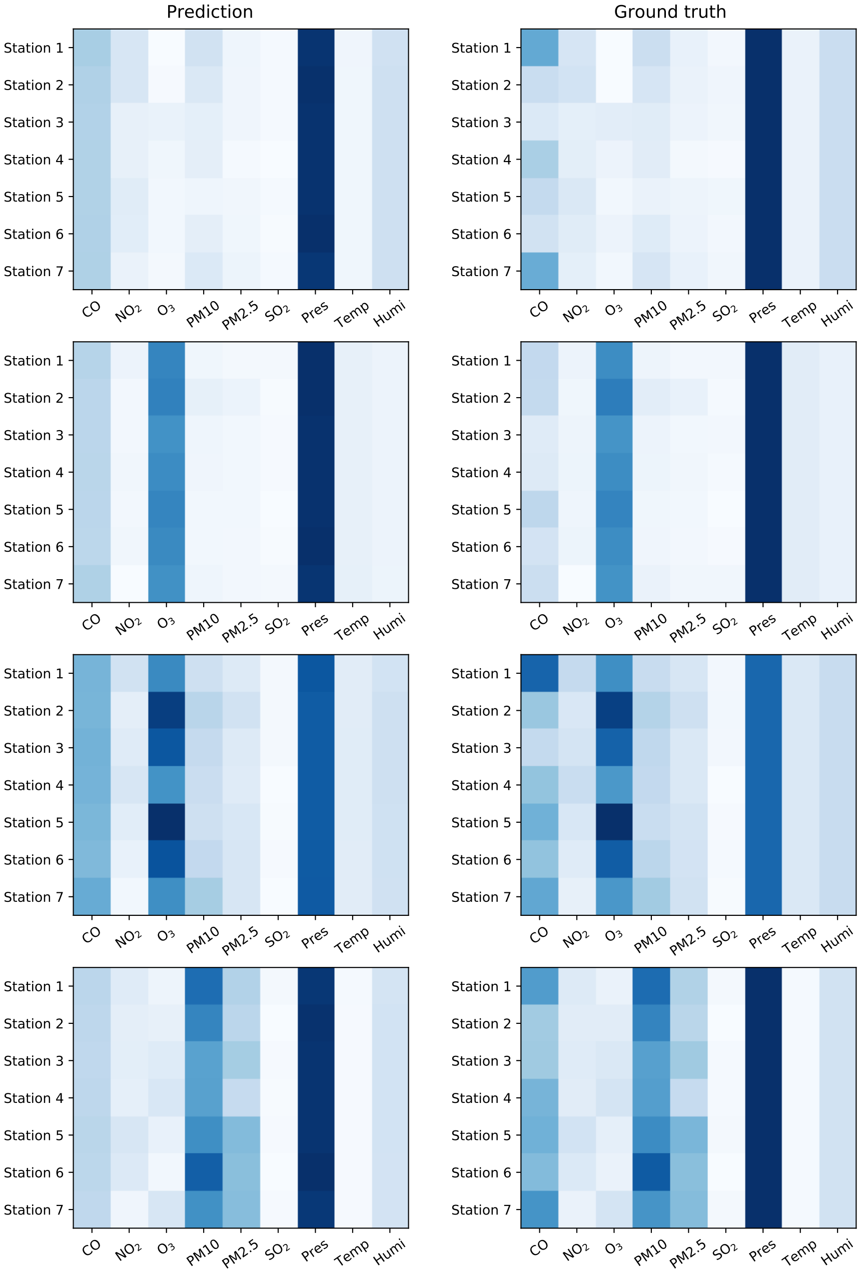


Figure 5. Visualization of the four ground truth *station* × *feature* frames randomly selected from the testing dataset (right panel) and the corresponding prediction (left panel) by the seq2seq model with time lag $l = 24$ and $\Delta_t = 48$. The color from dark blue to light blue corresponds to the values from large to small.

doi:10.20944/preprints202311.1091.v1

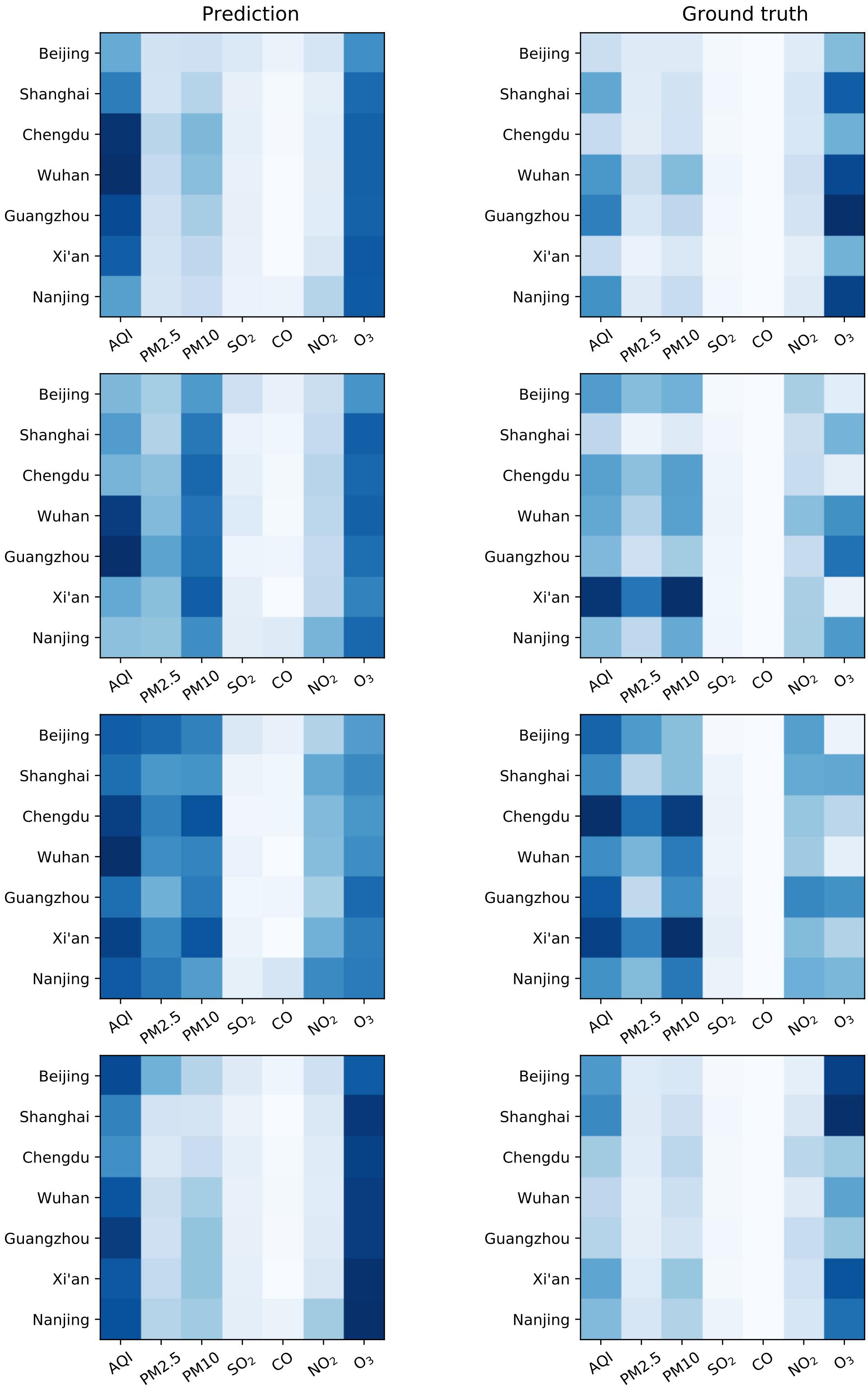


Figure 6. Visualization of the four ground truth *station* × *feature* frames randomly selected from the testing dataset (right panel) and the corresponding prediction (left panel) by the seq2seq model for seven major cities in China. The color from dark blue to light blue corresponds to the values from large to small.

4.5. Evaluation with three metrics

RMSE. As discussed in Section 3.3, the proposed model can predict the future level of air pollution from 1 hour to 48 hours by varying l from 1 to 48. Moreover, the data used for prediction can be a collection of samples in the last day or last several days, for which Δ_t is used to control the length of samples. In other words, we can vary the parameter Δ_t to obtain different training and testing datasets for different experiments. We explore six experiments by combinations of $l = \{1, 24, 48\}$ and $\Delta_t = \{24, 48\}$. Therefore, we utilize the data collected from yesterday and the last two days to predict the air quality for one hour later, one day later, and two days later, respectively.

Table 2 displays the RMSEs from the testing set for $l = \{1, 24, 48\}$ with respect to $\Delta_t = \{24, 48\}$. The RMSEs of the nonstationary LS-deGCN seq2frame and seq2seq models can be calculated directly based on the $station \times feature$ matrix. However, for the other three models, we need to conduct training and testing for each pollutant separately, and then calculate the average RMSE over all the pollutants. It is clear that our proposed models demonstrate significant improvements compared with the baseline methods and, in particular, the nonstationary LS-deGCN seq2seq model achieves the best performance under all the six scenarios. The LSTM-based methods outperform the traditional linear regression and support vector regression. Moreover, the RMSEs of our models increase as the time lag l increases and decrease as Δ_t takes a larger value. Intuitively, the prediction of air quality for two days later would be more challenging than that for one hour later, and the prediction would improve with more samples.

Accuracy. Table 3 compares the accuracy of air quality prediction among the five methods for combinations of $l = \{1, 24, 48\}$ and $\Delta_t = \{24, 48\}$. The proposed nonstationary LS-deGCN seq2frame and seq2seq models yield much higher accuracy than the others, and the nonstationary LS-deGCN seq2seq model achieves the highest accuracy among all.

MAE. Table 4 shows the MAE of all the methods for the six scenarios with $l = \{1, 24, 48\}$ and $\Delta_t = \{24, 48\}$. In those experiments, the two proposed models, the nonstationary LS-deGCN seq2frame and seq2seq models, report much lower MAE than the others, and the nonstationary LS-deGCN seq2seq model outperforms all the rest of competitors.

Table 2. Comparison of the root mean squared error (RMSE) among different methods based on the Chengdu testing dataset, where a smaller RMSE indicates a better result.

	Models	$l = 1$	$l = 24$	$l = 48$
Δ_t = 24	Linear regression	1.78	1.82	2.23
	Support vector regression	1.61	1.93	1..98
	LSTM seq2scalar	1.04	1.12	1.25
	Nonstationary LS-deGCN seq2frame	0.58	0.78	1.1
	Nonstationary LS-deGCN seq2seq	0.56	0.77	0.87
Δ_t = 48	Linear regression	1.67	1.79	2.03
	Support vector regression	1.52	1.76	1.95
	LSTM seq2scalar	0.87	0.92	0.95
	Nonstationary LS-deGCN seq2frame	0.45	0.62	0.67
	Nonstationary LS-deGCN seq2seq	**0.39**	**0.56**	**0.57**

Table 3. Comparison of accuracy among different methods based on the Chengdu testing dataset, where a higher value of accuracy indicates a better result.

	Models	$l = 1$	$l = 24$	$l = 48$
$\Delta_t = 24$	Linear regression	0.5634	0.5367	0.5278
	Support vector regression	0.5763	0.5598	0.5557
	LSTM seq2scalar	0.7021	0.7167	0.7198
	Nonstationary LS-deGCN seq2frame	0.7234	0.7545	0.7517
	Nonstationary LS-deGCN seq2seq	0.7365	0.7652	0.7482
$\Delta_t = 48$	Linear regression	0.5612	0.5423	0.5186
	Support vector regression	0.5834	0.5654	0.5521
	LSTM seq2scalar	0.6825	0.7212	0.7237
	Nonstationary LS-deGCN seq2frame	0.7235	0.7866	0.7655
	Nonstationary LS-deGCN seq2seq	**0.7655**	**0.8123**	**0.7785**

Table 4. Comparison of the mean absolute error (MAE) among different methods based on the Chengdu testing dataset, where a smaller MAE indicates a better result.

	Models	$l = 1$	$l = 24$	$l = 48$
$\Delta_t = 24$	Linear regression	0.0175	0.0211	0.0234
	Support vector regression	0.0158	0.0147	0.0186
	LSTM seq2scalar	0.0148	0.0167	0.0166
	Nonstationary LS-deGCN seq2frame	0.0092	0.0091	0.0101
	Nonstationary LS-deGCN seq2seq	0.0077	0.0089	0.0093
$\Delta_t = 48$	Linear regression	0.0178	0.0198	0.0211
	Support vector regression	0.0153	0.0132	0.0201
	LSTM seq2scalar	0.0136	0.0142	0.0154
	Nonstationary LS-deGCN seq2frame	0.0080	0.0079	0.0091
	Nonstationary LS-deGCN seq2seq	**0.0071**	**0.0078**	**0.0083**

For the three metrics, we also observe that the scenarios with $\Delta_t = 48$ outperform those with $\Delta_t = 24$, especially for the LSTM-based methods (i.e., LSTM seq2scalar, nonstationary LS-deGCN seq2frame and seq2seq). This is expected as prediction of air quality with the historical data of the last two days is a better strategy than that of only utilizing the data of yesterday.

For the results of RMSE and MAE in Table 2 and Table 4, we observe that the performances continue to decrease when the value of l increases. However, for the accuracy results in Table 3, we observe that the performance does not monotonically decrease with an increasing value of l for our models. In contrast, the accuracy achieves a peak at $l = 24$, and the performance deteriorates for both $l = 1$ and $l = 48$. This indicates that the three evaluation metrics do not exactly match with each other. As discussed earlier, the larger the time lag l, the more difficult it is to make a prediction. However, if l is too small, the overlapped information under the nonstationary LS-deGCN seq2seq model would be large.

5. Conclusions

In this paper, we train a non-stationary LS-deGCN to fit the spatial-temporal data, a multi-station air pollution dataset of Chengdu from January 1, 2013 to December 31, 2016. On one hand, the de-trending long-short term structure of LS-deGCN can capture the non-stationary temporal dependency between the predicted values and the historical records. On the other hand, the diffusion graph convolution on the $station \times features$ grid gains more power to extract the spatial dependency among stations than a fully connected network. According to the scheme of extracting samples and targets, we propose two sub-models, the non-stationary LS-deGCN seq2frame and non-stationary

LS-deGCN seq2seq. The target of the former model is a frame while the later is a l-shifted sequence. We evaluate both models based on the RMSE, accuracy, and MAE and demonstrate their outstanding performances in comparison with existing works. However, air quality prediction involves many other factors such as geographical information and traffic information. In the future work, we will incorporate other data and integrate them into the framework of the non-stationary LS-deGCN.

Author Contributions: Conceptualization, G.Y and B.L.; methodology, B.L and M.H; software, M.H.; validation, M.H.; formal analysis, B.L and M.H; investigation, M.H; resources, G.Y and B.L.; data curation, M.H; writing—original draft preparation, B.L and M.H.; writing—review and editing, G.Y; visualization, M.H.; supervision, G.Y and B.L.; project administration, G.Y and B.L; funding acquisition, G.Y. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: We thank the editors and reviewers for their helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

CNN	Convolutional neural network
LSTM	Long short-term memory
LS-deGCN	Long-short de-trending graph convolutional network
NLP	Natural language processing
RNN	Recurrent neural network
RMSE	Root mean squared error
MAE	Mean absolute error

References

1. Ayturan, Y.A.; Ayturan, Z.C.; Altun, H.O. Air pollution modelling with deep learning: A review. *International Journal of Environmental Pollution and Environmental Modelling* **2018**, *1*, 58–62.
2. Zhang, S.; Guo, B.; Dong, A.; He, J.; Xu, Z.; Chen, S.X. Cautionary tales on air-quality improvement in Beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2017**, *473*, 20170457.
3. Qi, Z.; Wang, T.; Song, G.; Hu, W.; Li, X.; Zhang, Z. Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality. *IEEE Transactions on Knowledge and Data Engineering* **2018**, *30*, 2285–2297.
4. Zhu, J.Y.; Sun, C.; Li, V.O. Granger causality based air quality estimation with spatio-temporal (ST) heterogeneous big data. In Proceedings of the 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2015, pp. 612–617.
5. Zhu, J.Y.; Sun, C.; Li, V.O. An extended spatio-temporal Granger causality model for air quality estimation with heterogeneous urban big data. *IEEE Transactions on Big Data* **2017**, *3*, 307–319.
6. Huang, C.J.; Kuo, P.H. A deep CNN-LSTM model for particulate matter (PM2. 5) forecasting in smart cities. *Sensors* **2018**, *18*, 2220. https://doi.org/https://doi:10.3390/s18072220.
7. Li, X.; Peng, L.; Yao, X.; Cui, S.; Hu, Y.; You, C.; Chi, T. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental Pollution* **2017**, *231*, 997–1004.
8. Fan, J.; Li, Q.; Hou, J.; Feng, X.; Karimian, H.; Lin, S. A spatiotemporal prediction framework for air pollution based on deep RNN. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2017**, *4*, 15–22.
9. Wang, J.; Song, G. A deep spatial-temporal ensemble model for air quality prediction. *Neurocomputing* **2018**, *314*, 198–206.
10. Shi, X.; Zhourong, C.; Hao, W.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 802–810.
11. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In Proceedings of the International Conference on Learning Representations (ICLR '18), 2018.
12. Deodatis, G. Non-stationary stochastic vector processes: seismic ground motion applications. *Probabilistic Engineering Mechanics* **1996**, *11*, 149–167.

13. Wang, Y.; Zhang, J.; Zhu, H.; Long, M.; Wang, J.; Yu, P.S. Memory In Memory: A predictive neural network for learning higher-order nonstationarity from Spatio-temporal dynamics. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9154–9162.
14. Liu, B.; Yan, S.; Li, J.; Li, Y. Forecasting PM2.5 concentration using spatio-temporal extreme learning machine. In Proceedings of the Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on. IEEE, 2016, pp. 950–953.
15. Ghaemi, Z.; Alimohammadi, A.; Farnaghi, M. LaSVM-based big data learning system for dynamic prediction of air pollution in Tehran. *Environmental Monitoring and Assessment* **2018**, *190*, 300.
16. Liu, B.C.; Binaykia, A.; Chang, P.C.; Tiwari, M.K.; Tsao, C.C. Urban air quality forecasting based on multi-dimensional collaborative support vector regression (svr): A case study of Beijing-Tianjin-Shijiazhuang. *PLOS One* **2017**, *12*.
17. Mukhopadhyay, S.; Sahu, S.K. A Bayesian spatiotemporal model to estimate long-term exposure to outdoor air pollution at coarser administrative geographies in England and Wales. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **2018**, *181*, 465–486.
18. Russo, A.; Raischel, F.; Lind, P.G. Air quality prediction using optimal neural networks with stochastic variables. *Atmospheric Environment* **2013**, *79*, 822–830.
19. Biancofiore, F.; Busilacchio, M.; Verdecchia, M.; Tomassetti, B.; Aruffo, E.; Bianco, S.; Di Tommaso, S.; Colangeli, C.; Rosatelli, G.; Di Carlo, P. Recursive neural network model for analysis and forecast of PM10 and PM2. 5. *Atmospheric Pollution Research* **2017**, *8*, 652–659.
20. Kök, İ.; Şimşek, M.U.; Özdemir, S. A deep learning model for air quality prediction in smart cities. In Proceedings of the IEEE International Conference on Big Data (Big Data). IEEE, 2017, pp. 1983–1990.
21. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Computation* **1997**, *9*, 1735–1780.
22. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
23. Guo, T.; Lin, T.; Lu, Y. An interpretable LSTM neural network for autoregressive exogenous model. In Proceedings of the Workshop of International Conference on Learning Representations, 2018.
24. Xu, J.; Liu, X.; Wilson, T.; Tan, P.N.; Hatami, P.; Luo, L. MUSCAT: Multi-scale spatio-temporal learning with application to climate modeling. In Proceedings of the International Joint Conference on Artificial Intelligence, 2018, pp. 2912–2918.
25. Wilson, T.; Tan, P.N.; Luo, L. A low rank weighted graph convolutional approach to weather prediction. In Proceedings of the IEEE International Conference on Data Mining. IEEE, 2018, pp. 627–636.
26. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In Proceedings of the International Conference on Learning Representations, 2018.
27. Srivastava, N.; Mansimov, E.; Salakhudinov, R. Unsupervised learning of video representations using LSTMs. In Proceedings of the International Conference on Machine Learning, 2015, pp. 843–852.
28. Shuman, D.I.; Narang, S.K.; Frossard, P.; Ortega, A.; Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine* **2013**, *30*, 83–98.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.