# Exploiting Spatiotemporal Patterns for Accurate Air Quality Forecasting using Deep Learning

Yijun Lin, Nikhit Mago, Yu Gao
Spatial Sciences Institute
University of Southern California
[yijunlin,mago,gaoyu]@usc.edu

Yaguang Li
Department of Computer Science
University of Southern California
yaguang@usc.edu

Yao-Yi Chiang
Spatial Sciences Institute
University of Southern California
yaoyic@usc.edu

Cyrus Shahabi
Department of Computer Science
University of Southern California
shahabi@usc.edu

José Luis Ambite
Information Sciences Institute
University of Southern California
ambite@isi.edu

## ABSTRACT

Forecasting spatially correlated time series data is challenging because of the linear and non-linear dependencies in the temporal and spatial dimensions. Air quality forecasting is one canonical example of such tasks. Existing work, e.g., auto-regressive integrated moving average (ARIMA) and artificial neural network (ANN), either fails to model the non-linear temporal dependency or cannot effectively consider spatial relationships between multiple spatial time series data. In this paper, we present an approach for forecasting short-term $PM_{2.5}$ concentrations using a deep learning model, the geo-context based diffusion convolutional recurrent neural network, GC-DCRNN. The model describes the spatial relationship by constructing a graph based on the similarity of the built environment between the locations of air quality sensors. The similarity is computed using the surrounding "important" geographic features regarding their impacts to air quality for each location (e.g., the area size of parks within a 1000-meter buffer, the number of factories within a 500-meter buffer). Also, the model captures the temporal dependency leveraging the sequence to sequence encoder-decoder architecture. We evaluate our model on two real-world air quality datasets and observe consistent improvement of 5%-10% over baseline approaches.

## CCS CONCEPTS

• **Information systems** → *Spatial-temporal systems*;

## KEYWORDS

Air Quality Forecasting, Spatiotemporal Time Series Analysis, $PM_{2.5}$, Deep Learning

## 1 INTRODUCTION

Fine particulate matter ($PM_{2.5}$) consists of particles with aerodynamic diameters less than 2.5 $\mu$m. Typically, moving vehicle exhausts, industrial emissions, and burning sources (e.g., coal combustion and wildfiles) are the primary contributors to generate fine particulate matters. Many epidemiological studies [23, 27] have shown that exposure to $PM_{2.5}$ is strongly associated with various health effects. Long-term (chronic) exposure may lead to deterioration of the respiratory system [11], cause damage to the body's immune system and increase the risk of cardiovascular diseases [15]. Short-term (acute) exposure also raises acute health concerns such as eye irritation and breathing difficulty. Sensitive groups like the elderly, children and people with lung and heart diseases are more vulnerable to air pollution-related diseases, such as children asthma attack [21]. Besides, fine particulate matter and its derivatives also cause adverse effects on the environment such as poor visibility, global climate change [29], and ecological damage [33].

In 2006, the World Health Organization (WHO) suggested that long-term exposures (an annual average of $PM_{2.5}$ concentrations) should not exceed $10mg/m^3$ and short-term exposures (24-hour average of $PM_{2.5}$ concentrations) should not exceed $25mg/m^3$ [22]. Many countries have built air quality monitoring stations to report the concentrations of major pollutants such as sulfur dioxide ($SO_2$), nitrogen dioxide ($NO_2$), and particulate matters ($PM_{2.5}$ and $PM_{10}$). In the United States, the Environmental Protection Agency (US EPA) has established the ambient air monitoring network, which provides hourly measurements for various pollutants (e.g., $PM_{2.5}$ and $PM_{10}$). US EPA also set a national air quality standard, the air quality index (AQI), to indicate health risk level of the pollution level. US EPA has defined the AQI values for five major air pollutants:[1] the ground-level ozone, particle pollution, carbon monoxide, sulfur dioxide, and nitrogen dioxide. The AQI consists of six categories: "Good", "Moderate", "Unhealthy for Sensitive Groups", "Unhealthy", "Very Unhealthy", and "Hazardous" with a range from 0 to 500. One can convert the concentration value to the corresponding AQI and its health risk category. For example, a 50 $\mu g/m^3$ $PM_{2.5}$ measurement corresponds to an AQI value of 12 and is in the "Good" category, which means "*It's a great day to be active outside*" according to the EPA definition. Generally, an AQI value below 100 is an acceptable range. When the AQI value is growing to 100, it is considered to be

*SIGSPATIAL '18, November 6–9, 2018, Seattle, WA, USA*
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5889-7/18/11…$15.00
https://doi.org/10.1145/3274895.3274907

[1] https://airnow.gov/index.cfm?action=aqibasics.aqi

unhealthy for certain sensitive groups and the public need to take protective actions.[2]

Besides establishing monitoring stations to report the real-time air quality status, there is an increasing demand to forecast the air quality pollutants, which not only supports governments to make policies in pollution control but also informs the general public to take advanced actions like staying at home. However, forecasting air quality is a challenging task. First, air quality values can vary significantly over time and across locations. Figure 1 shows the $PM_{2.5}$ AQIs at two reporting areas in Los Angeles, “Central LA CO” and “Santa Clarita Vly”, during the period of early January and July. We have observed that “Central LA CO” generally shows higher $PM_{2.5}$ AQI values than that in “Santa Clarita Vly” (could be the heavy traffic congestion in “Central LA CO”). Besides, the $PM_{2.5}$ AQI values are significantly higher in July than January (could be the holiday events in July). Second, sudden changes in the observations make it challenging for a general model to capture the temporal patterns and to forecast the future values. Such abrupt changes might be caused by some unusual situations, such as strong winds, raining, events (e.g., fireworks), and sudden factory emissions. Figure 1 shows extremely high $PM_{2.5}$ AQI values during the Independence Day Holiday (from July $4^{th}$ to $6^{th}$) that holds many celebration activities (e.g., fireworks). Third, air quality is influenced by various complex factors, such as meteorological effects, surrounding land usages, and the chemical processes of air pollutants. These challenges together with the fact that air quality monitoring stations are usually sparse make it difficult to forecast air quality values.

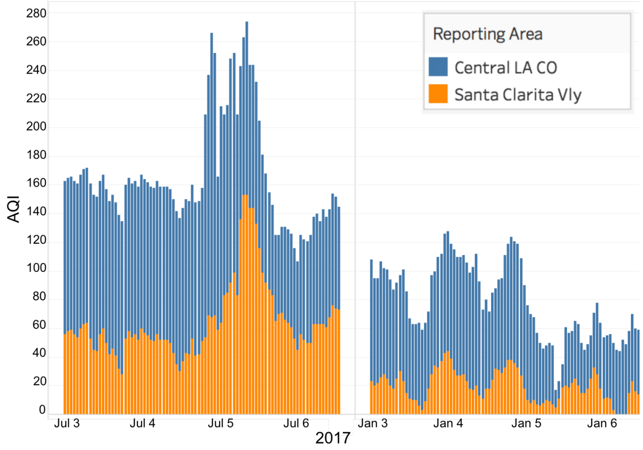


**Figure 1: An example of $PM_{2.5}$ AQIs at two reporting areas in Los Angeles in early July (left) and January (right) 2017**

An abundant existing work takes advantage of the observation data from air quality monitoring stations to build and validate real-time air quality forecasting (RT-AQF) models [6, 10, 26] for computing future air quality status in a short term (1-5 days). Various time series forecasting methods have been applied to the air quality forecasting problem. The commonly used methods include auto-regressive integrated moving average (ARIMA) [14], Kalman filtering (KF) [9], regression method [5], and artificial neural network (ANN) [25]. ARIMA and KF only work for stationary time series data, which fail to capture the dynamic trends in the air quality data, while regression methods like Linear Regression are incapable of handling the non-linearity in time series data. ANN based methods [7, 20, 24, 25] have been used to solve RT-AQF problems for various air pollutants. Though ANN is able to capture non-linear temporal patterns in time series data, it fails to handle spatial dependency in location-dependent time series data [20, 24].

A general way to define the spatial dependency is using geographic distance [34], i.e., if a sensor location is close to the target location, the sensor is considered as the neighbor that will contribute largely for forecasting the air quality at the target location. However, when the monitoring stations are sparse or far apart from each other, the geographic distance becomes less informative as all distances are similarly high values. Therefore, geographic distance cannot fully capture the similarity in environmental characteristics (like near industrial areas or green lands) that have various impacts on air quality. In [34], the authors proposed a hybrid forecasting model to separately deal with temporal and spatial correlations in air quality measurements. The hybrid forecasting model includes a temporal predictor (linear regression) to model the air quality trend at a target station, a spatial predictor to model the impact of meteorological data and air quality readings from other stations, an aggregator to integrate the results from the previous two predictors, and an inflection predictor to detect the sudden drops. Though the proposed forecasting model demonstrated promising performance, it uses fixed-distance (geographic distance) buffers to capture neighborhoods, which might not be able to represent spatial dependency sufficiently [19]. Also, it uses separate models for correlating temporal and spatial patterns, which might not comprehensively represent the spatiotemporal patterns as a whole. It remains a challenge to build a general air quality forecasting model with a universe mean to define the spatial dependency for air quality in location-dependent time-series data and handle the spatial and temporal dependencies jointly.

In this paper, we propose a novel deep learning model, the geo-context based diffusion convolutional recurrent neural network (GC-DCRNN), to forecast the $PM_{2.5}$ concentrations in the next several hours (e.g., 1, 6, 12, 18, and 24 hours) at a given location. In our method, we utilize the neighborhood characteristics to represent the spatial correlation, which means two locations would have similar air quality conditions if they share a similar built environment. We take advantage of our previous work on air quality prediction [19] to automatically select the “important” geographic feature types (e.g., factories within 1,000 meters) that have a significant impact on air quality at a given location. We compute the similarity of the “important” geographic feature types around sensors and construct a graph in which the nodes are the sensors and edge weights are the similarities (section 3). Then we apply the pre-constructed graph in the diffusion convolutional recurrent neural network (DCRNN) [18] to build an air quality forecasting model (GC-DCRNN). The inputs of the GC-DCRNN model are the geo-context based graph, a sequence of air quality readings, and a sequence of meteorological data (e.g., humidity, temperature, and wind speed) over the past few hours (e.g., 24 hours) at all the stations. The outputs are

[2] https://airnow.gov/aqi/aqi-basics

the forecasted $PM_{2.5}$ AQIs/concentrations, i.e., a sequence of forecasting values in next 24 hours at all the stations. In general, the GC-DCRNN model utilizes the diffusion convolutional operation on the geo-context based graph to capture the spatial dependency. To jointly model the spatial and temporal dependencies in the air quality data, the model further integrates the diffusion convolutional operation into the recurrent neural network (section 3). We test our model on two real-world air quality datasets in Los Angeles and Beijing and compare our forecasting results to other existing approaches. The experiments show our GC-DCRNN model consistently outperforms other air quality forecasting methods (section 4).

The main contribution of this paper is that we present the GC-DCRNN model that jointly manipulates the spatial and temporal dependencies in location-dependent air quality time series data for forecasting. We utilize an automatic approach to describe the spatial dependency by considering the similarity of the built environment with regard to air quality. We construct a geo-context based graph that enables the DCRNN model to handle the spatial correlations by automatically selecting important geographic feature types that have a significant impact on air quality pollutants. Our model improves the traditional air quality models that overlook the spatial dependency. The forecasting results from our model are particularly important in the study of air pollution and can help with the analysis of air pollution-related diseases, such as predicting children asthma attack.

The rest of this paper is organized into four sections. Section 2 presents an introduction to the data sources. Section 3 describes the methodology of the deep learning model for forecasting $PM_{2.5}$ concentrations. Section 4 discusses the related work on air quality forecasting and time series forecasting. Section 5 presents the experiments and evaluation of the results. Finally, Section 6 concludes the paper with a discussion of future work.

## 2 DATA SOURCES

In general, our forecasting model utilizes the following datasets: 1) the $PM_{2.5}$ time series data, which are for training and validating the model, 2) the meteorological time series data, which serve as auxiliary features in the model, 3) the geographic data for building a graph that denotes the spatial relationship between monitoring stations.

### 2.1 AQS (Air Quality System) Data

*2.1.1 Los Angeles air quality data from EPA.* We collect the Los Angeles air quality data, including the $PM_{2.5}$, $PM_{10}$, and $O_3$ AQI observations, every hour through the EPA's Airnow web service[3] using multiple zip codes. A total of 13 reporting areas are providing $PM_{2.5}$ AQI observations in the Los Angeles Metropolitan Area covering an area of approximate 4,751 mi$^2$ (Figure 2). Table 1 gives an example of the structured $PM_{2.5}$ AQIs in one of the reporting areas, Central LA CO.

*2.1.2 Beijing air quality data from Biendata.* Biendata provides publicly accessible air quality data of Beijing in the KDD CUP of Fresh Air.[4] It contains hourly air quality concentrations for various pollutants in Beijing including $PM_{2.5}$, $PM_{10}$, $NO_2$, CO, $O_3$, and $SO_2$. As Figure 3 shows, there are a total 35 monitoring stations in Beijing, with the approximate area size of 6,490 mi$^2$.

**Table 1: Example of $PM_{2.5}$ AQIs in Central LA CO**

| Monitoring Station | Timestamp | $PM_{2.5}$ AQI |
|---|---|---|
| Central LA CO | 2017-03-04 12:00:00 | 50 |
| Central LA CO | 2017-03-04 13:00:00 | 53 |
| Central LA CO | 2017-03-04 14:00:00 | 55 |
| Central LA CO | 2017-03-04 15:00:00 | 58 |
| Central LA CO | 2017-03-04 16:00:00 | 60 |

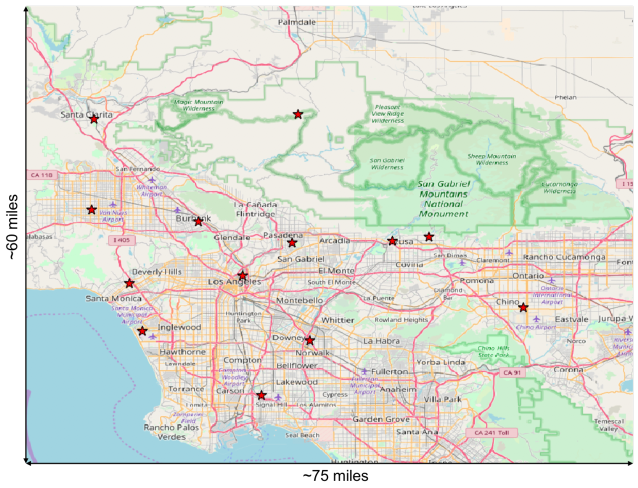


**Figure 2: The locations of Reporting Areas that provide $PM_{2.5}$ AQI observations in Los Angeles. The approximate geographic size of the displayed map is 75×60 mi$^2$.**

### 2.2 Meteorological Data

We collect meteorological data through the Dark Sky API.[5] Dark Sky reports fine-scale weather data (including temperature, humidity, wind speed, wind direction, etc.) all over the world. For each reporting area or monitoring station in Los Angeles and Beijing, we query hourly meteorological data based on the coordinates of each location. The meteorological data have the same time resolutions (i.e., hourly) as the air quality data.

### 2.3 Geographic Data

OpenStreetMap (OSM) is the crowd sourced world map. It provides a variety of geographic features, e.g., land uses, roads, water areas, and buildings.[6] For example, the geographic feature type "road", represented as line, contains many subtypes such as motorway and pedestrian roads. "Land use" describes the function of an area, such

[3] https://docs.airnowapi.org/webservices

[4] https://biendata.com/competition/kdd_2018/data/

[5] https://darksky.net/dev/docs

[6] http://wiki.openstreetmap.org/wiki/Map_Features

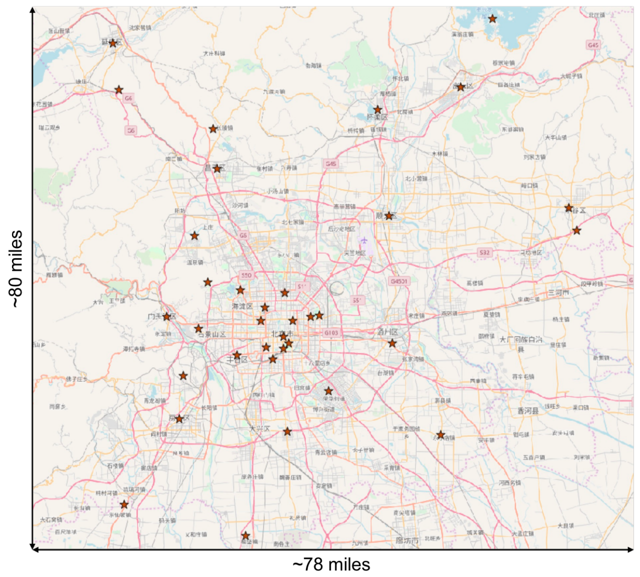


**Figure 3: The locations of monitoring stations in Beijing, China. The approximate geographic size of the displayed map is 78×80 mi$^2$.**

as industrial areas and commercial areas. “Water area” represents the area of a lake or pond. In this paper, our approach computes the values of the geographic feature types within various buffers around the sensor locations utilizing OpenStreetMap data.

# 3 METHODOLOGY

Our goal is to forecast the PM$_{2.5}$ values in next 24 hours at a given location. We construct a graph to represent the spatial relationship in the sensor network. We define an indirect graph G = (V, E, A) to represent the network, where V is a set of sensor nodes, E is a set of edges that link the sensors, and A is a weighted adjacency matrix representing the nodes proximity (e.g., the geographic similarity). Denote the observed data on G as a graph signal $\boldsymbol{X} \in \mathbb{R}^{N\times P}$, N is the number of nodes in the graph and P is the number of total features on each node. $\boldsymbol{X}$ consists of two parts, the air quality readings (denoted as $\boldsymbol{X_R} \in \mathbb{R}^{N\times P_R}$) and auxiliary features (denoted as $\boldsymbol{X_A} \in \mathbb{R}^{N\times P_A}$). Note that $P = P_R + P_A$. Let $\boldsymbol{X}^{(t)}$ represents the graph signal at time $t$, $T'$ represents the number of previous hours, i.e., from $(t + T' - 1)$ to $(t)$, and $T$ represents the number of future hours, i.e., from $(t + 1)$ to $(t + T)$. The model aims to learn a function $h$ that maps $T'$ (historical graph signals) to future $T$ graph signals, given a graph G:

$$[\boldsymbol{X}^{(t+T'-1)}, \cdots, \boldsymbol{X}^{(t)}; G] \xrightarrow{h} [\boldsymbol{X_R}^{(t+1)}, \cdots, \boldsymbol{X_R}^{(t+T)}]$$

## 3.1 Data Preprocessing

In practice, time series data or streaming data are generally incomplete and contain noise (or outliers). Exceptional abruptions of the monitoring sensors might cause missing values in air quality data and meteorological data. The missing values could have a large impact on the performance of the analytic models. Therefore, it is essential to eliminate the missing values in the data.

Our approach computes the mean value in a six-hour sliding window to replace the missing values. For example, assume we have an hourly time series of PM$_{2.5}$ AQIs, [... , 18, 23, 27, null, 30, 25, 22, ...]. By applying a six-hour sliding window, we are able to fill up the null value with the mean of the window [18, 23, 27, null, 30, 25, 22], that is 24. The data preprocessing step of imputing missing values is only performed on the training data and the testing data are untouched.

## 3.2 Graph Construction

In this section, we compute the similarity of the “important” geographic features around monitoring stations to represent their spatial correlation and construct the graph for diffusion convolution in the next step. We take advantage of our previous work [19], which uses a data-driven method to automatically select critical geographic features that have a significant impact on air quality.

### *3.2.1 Grouping Stations on PM$_{2.5}$ Concentrations.*

Our approach first identifies the monitoring stations that have similar PM$_{2.5}$ AQI temporal patterns. For example, the monitoring stations near industrial areas always show a higher PM$_{2.5}$ value than that in mountain areas, so they should be grouped into separate clusters. However, traditional cluster methods like K-means do not work for high dimensional time series data [2]. Due to the high dimension of our air quality data (more than 7,000, each hour is a dimension), we utilize the piece-wise aggregate approximation (PAA) [12] to reduce the dimension by representing the original sequences with the average of daily highest three values and the average of daily lowest three values. Then we use K-means to group the lower dimensional data (less than 600) after applying PAA with the K at the elbow point to get the clustering label for each location.

### *3.2.2 Constructing Geographic Abstraction.*

Our approach computes an aggregated value for each geographic feature type within various buffers (e.g., 100 meters and 3,000 meters) around each monitoring station using OpenStreetMap data and uses the values to construct a feature vector. Figure 4 is an example showing how we construct the geographic abstraction vector. There are roads, land uses, and buildings around the monitoring station, A, with the 100-meter and 200-meter buffers. For polygon features, such as land uses, we compute the sum of areas for various types (OSM types) within some buffers. Monitoring station A has 500 m$^2$ water areas within a 100-meter buffer while 950 m$^2$ water areas and 740 m$^2$ green lands within a 200-meter buffer, which generates the geographic abstraction vector as [500, 0, 950, 740]. For the line features, like roads and aeroways, we take the sum of lengths of various feature types as the geographic abstraction. Monitoring station A has 23-meter pedestrian and 30-meter motorway within a 100-meter buffer while 43-meter Pedestrian and 200m Motorway within a 200-meter buffer. Thus, the geographic abstraction vector is constructed as [23, 30, 43, 200]. For point features, like buildings, our approach counts the number of various feature types. For example, monitoring station A has 2 houses and 2 commercial buildings within a 100-meter buffer while 6 houses and 5 commercial buildings within a 200-meter buffer, which corresponds to the geographic abstraction vector as [2, 2, 6, 5]. The entire feature vector of geographic abstraction for monitoring station A can be represented as [500, 0, 950, 740, 23, 30, 43, 200, 2, 2, 6, 5]. In practice, we create buffers from 100 meters to 3,000 meters with an interval of 100 meters and generate the value for each unique geographic feature type with the buffers.

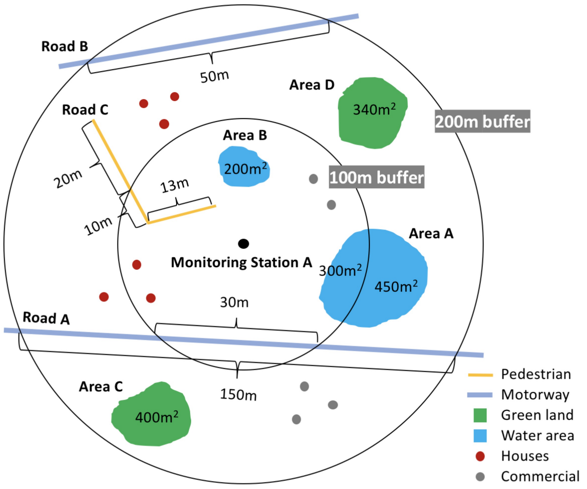


**Figure 4: Examples of the geographic features (i.e., roads, land uses, and buildings) and the various geographic feature types in the 100-meter and 200-meter buffers**

### 3.2.3 *Computing Geo-Context.*

Our approach utilizes the clustering labels and the geographic abstraction as the inputs of a random forest classifier to automatically quantify the importance of individual components in the geographic abstraction vector. The random forest classifier identifies the important geographic feature types and a buffer size that have the most impact on $PM_{2.5}$ concentrations. For unimportant features, the model gives the importance as zero. Our approach keeps those important feature components (non-zeroes) to form a new feature vector, called the geo-context. To construct the graph, we compute the similarity between the geo-context of the monitoring stations as the edge weight by using the Euclidean distance. In the next step, we embed the graph in the DCRNN model to handle the spatial dependency.

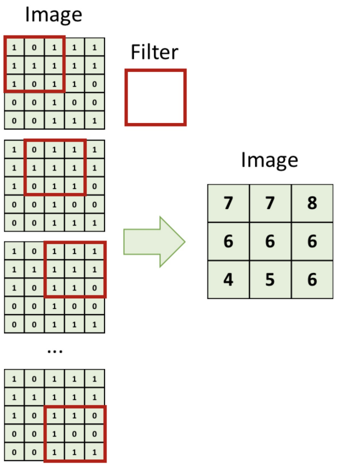
## 3.3 Diffusion Convolution

In this section, we describe the diffusion convolution process for the constructed graph. Traditional convolutional neural network (CNN) works for the grid-structured data, e.g., image. The convolution operation scans across the image with a filter to extract the features. For example, in Figure 5, suppose the $3 \times 3$ filter is defined to add all the elements within the filter, we can map a $5 \times 5$ input image to a $3 \times 3$ output image whose elements are the sum of each $3 \times 3$ grids based on the filter. The diffusion convolution extends this idea to the general graph-structured data. DCRNN [18] defines diffusion convolution as the combination of diffusion processes with different steps on the graph. Specifically, the k diffusion step represents the "distance" to the center point (i.e., the current forecasting location). Figure 6 shows an example of diffusion process, at each step i, the model looks at the neighbors that are k-step away from the center point and computes the transition matrices for this step.[7] The diffusion convolutional filter then adds the transition matrices with some probability $\theta$, which is learned during the training step. Formally, the *diffusion convolution operation* $\star_G$ over a graph signal $\boldsymbol{X} \in \mathbb{R}^{N \times P}$ and a filter $f_\theta$ is defined as:

$$\boldsymbol{X}_{:,p} \star_G \boldsymbol{f_\theta} = \sum_{k=0}^{K-1} (\theta_k (\boldsymbol{D}^{-1}\boldsymbol{A})^k) \boldsymbol{X}_{:,p} \qquad (1)$$

where $\boldsymbol{\theta} \in \mathbb{R}^{K \times 2}$ are the parameters for the filter, $D$ denotes the diagonal degree matrix of the graph and $\boldsymbol{D}^{-1}\boldsymbol{A}$ represents the transition matrices of the diffusion process.

The diffusion convolutional layer is defined with the definition of diffusion convolutional operation in Equation 1. It maps P-dimensional features to Q-dimensional output, which is similar to extracting features in CNN.



|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |

|   |   |   |
|---|---|---|
| 7 | 7 | 8 |
| 6 | 6 | 6 |
| 4 | 5 | 6 |

**Figure 5: An example of Convolutional Neural Networks**

## 3.4 DCRNN

Besides spatial dependency, the model addresses temporal dependency with a recurrent neural network (RNN) model, which is commonly used for handling sequential data. The basic idea of using RNN for time series data analysis is that it not only considers the current input as in the traditional machine learning algorithms, but also makes use of the information from the previous time point. For example, in Figure 7, suppose $\boldsymbol{X}^{(t)}$ is the input at time $t$ and $\boldsymbol{H}^{(t)}$ is the hidden state at time $t$, which is the memory cell in the network. $\boldsymbol{H}^{(t)}$ is obtained from both the previous hidden cell $\boldsymbol{H}^{(t-1)}$

[7] Reprinted from the poster of [18] with permission from the corresponding author.

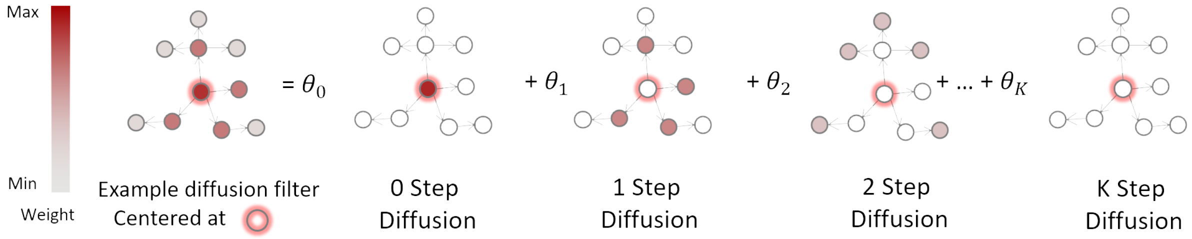


**Figure 6: Example Diffusion Process**

and current input $\boldsymbol{X}^{(t)}$, as $\boldsymbol{H}^{(t)} = f(\boldsymbol{W}\boldsymbol{X}^{(t)} + \boldsymbol{U}\boldsymbol{H}^{(t-1)})$ where $f$ is a nonlinear variation function like *tanh* and *relu*.

DCRNN leverages the gated recurrent units (GRU) [4] as the cell, which is a simple yet powerful variant of RNN. As Figure 8 shows, the GRU structure contains two gates: the reset gate and update gate. The reset gate is used to decide if $\boldsymbol{H}_{t-1}$ will pass information to $\boldsymbol{H}_t$. The update gate is used to decide how much information of $\boldsymbol{H}_{t-1}$ will give to $\boldsymbol{H}_t$. Generally, GRU first defines the gate signals based on the input ($\boldsymbol{X}_t$ and $\boldsymbol{H}_{t-1}$) with the following formula:

$$\boldsymbol{r}^{(t)} = \sigma(\boldsymbol{W}_r\boldsymbol{X}^{(t)} + \boldsymbol{U}_r\boldsymbol{H}^{(t-1)} + \boldsymbol{b}_r)$$
$$\boldsymbol{u}^{(t)} = \sigma(\boldsymbol{W}_u\boldsymbol{X}^{(t)} + \boldsymbol{U}_u\boldsymbol{H}^{(t-1)} + \boldsymbol{b}_u)$$

where $\boldsymbol{r}^{(t)}$ is the reset gate and $\boldsymbol{u}^{(t)}$ is the update gate at time t. $\boldsymbol{W}_r$, $\boldsymbol{U}_r$, $\boldsymbol{W}_u$, and $\boldsymbol{U}_u$ are the parameters for corresponding gate. The variable $b_r$ and $b_u$ are the bias.

After getting the gate signals, the previous hidden status $\boldsymbol{H}_{t-1}$ is "reset" through the reset gate and combined with $\boldsymbol{X}_t$ with the following formula:

$$\boldsymbol{C}^{(t)} = \tanh(\boldsymbol{W}_c\boldsymbol{X}^{(t)} + \boldsymbol{U}_c(\boldsymbol{r}^{(t)} \odot \boldsymbol{H}^{(t-1)}) + \boldsymbol{b}_c)$$

where $\boldsymbol{W}_c$ and $\boldsymbol{U}_c$ are the parameters. The variable $\boldsymbol{b}_c$ is the bias. The operation $\odot$ is the Hadamard product, which multiplies the elements on the corresponding locations in the two matrices.

Then the update gate chooses to keep or ignore the information to achieve the new hidden status at time t.

$$\boldsymbol{H}^{(t)} = \boldsymbol{u}^{(t)} \odot \boldsymbol{H}^{(t-1)} + (1 - \boldsymbol{u}^{(t)}) \odot \boldsymbol{C}^{(t)}$$

In order to capture the spatiotemporal dependencies, DCRNN replaces the matrix multiplication in GRU with the diffusion convolution as following:

$$\boldsymbol{r}^{(t)} = \sigma(\Theta_r \star_G [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b}_r)$$
$$\boldsymbol{u}^{(t)} = \sigma(\Theta_u \star_G [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b}_u)$$
$$\boldsymbol{C}^{(t)} = \tanh(\Theta_C \star_G [\boldsymbol{X}^{(t)}, (\boldsymbol{r}^{(t)} \odot \boldsymbol{H}^{(t-1)})] + \boldsymbol{b}_c)$$
$$\boldsymbol{H}^{(t)} = \boldsymbol{u}^{(t)} \odot \boldsymbol{H}^{(t-1)} + (1 - \boldsymbol{u}^{(t)}) \odot \boldsymbol{C}^{(t)}$$

where $\star_G$ denotes the *diffusion convolution* defined in Equation 1 and $\Theta_r$,$\Theta_u$,$\Theta_C$ are parameters for the corresponding filters.

To conduct the multi-step ahead forecasting (i.e., in next 1-24 hours), DCRNN utilizes the *Sequence to Sequence* architecture [28]. Precisely, during the training step, sub-sequences of the historical time series are fed into the encoder. For example, a vector of 6-hour $PM_{2.5}$ AQIs as [13, 14, 16, 21, 20, 19] is put in the encoder. The decoder takes the final states of the encoder as initialization and emits the corresponding result as a sequence, which is fed with given ground truth observations, i.e., the actual $PM_{2.5}$ AQI values for the next 6 hours. During the testing step, the model generates forecasting results, which are compared with ground truth to evaluate the model. In this way, the model can generate air quality forecasting results given data from the previous hours by handling both spatial and temporal dependency simultaneously.

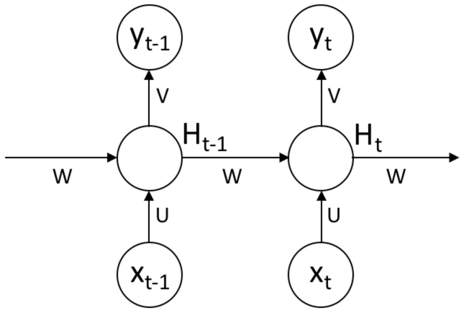


**Figure 7: An example of Recurrent Neural Networks**

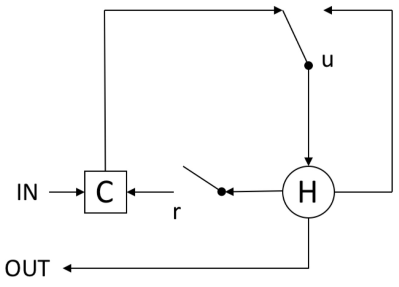


**Figure 8: Gated Recurrent Unit Structure**

## 4 EXPERIMENTS AND RESULTS

We utilized the air quality data, meteorological data, and OpenStreetMap data for building the air quality forecasting model and evaluating our model on two real-world datasets (Los Angeles and

Beijing) described in Section 2. We conducted the experiment in a Docker container deployed on the GPU server with four physical cores and 64GB memory for the deep learning model. The model was implemented with Python 2.7 and the Tensorflow [1] framework. The baseline models were executed on CPU only with 8GB memory. All geospatial computing was done in PostGIS.

## 4.1 Environmental Settings

In the experiment, we utilized the $PM_{2.5}$ observations from 2017-01-01 00:00:00 to 2018-03-01 00:00:00 for training and testing our air quality modeling approach with Los Angeles and Beijing datasets. We split the air quality data as well as the meteorological Data into training data (from 2017-01-01 to 2017-12-31) and testing data (from 2018-01-1 to 2018-03-01). We performed some data preprocessing for the air quality data and the meteorological data. For the Los Angeles dataset, we removed four reporting areas (i.e., *E San Gabriel V-1*, *NW Coastal LA*, *SW Coastal LA*, and *E San Fernando Vly*) whose air quality data are identical to some other areas. Note that the Beijing dataset includes nearly four times more available air quality reporting stations than the Los Angeles dataset (35 vs. 9).

We used the previous 24-hour observations (i.e., sequence length = 24) to forecast the next 24-hour $PM_{2.5}$ values (i.e., horizon = 24) for evaluating the performance of our air quality forecasting model. Suppose $\boldsymbol{Y} = [y_1, y_2, \cdots, y_n]$ represents the ground truth and $\widehat{\boldsymbol{Y}} = [\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n]$ represents the forecasting values, where n denotes the indices of the observed samples. We evaluated the model by comparing our results with a number of baseline methods by using the following metrics and missing values were excluded in calculating these metrics:

1. Mean absolute error (MAE) is simply the summation of difference between two corresponding variables divided by the total number of observations:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

2. Root Mean Squared Error (RMSE) was calculated in a similar fashion as MAE given by the formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}}$$

## 4.2 Baseline Methods

*4.2.1 Linear Regression (LR).* LR can capture linear dependencies between the input variables (e.g., features) and the output variables (e.g., predictions). These predictions are auto-correlated, for example $X_{t+1}$ usually depends on $X_t$. Linear Regression studies the relationships between the variables and works well with data that follow a linear trend. For time series analysis, LR captures the autocorrelation of variables and with some lag variables emulates Autoregression, a time series analysis technique. For example, we can forecast 24 hours ahead based on the previous 24-hour data. Here the number of lag variables or sequence length is equal to 24 and horizon is equal to 24.

*4.2.2 Vector Autoregression Regression (VAR).* VAR is similar to LR but it considers multiple time series data to predict the independent variable for each time series. For example, instead of using only one time series data from one sensor, all time series from other sensors are fed into the model. The intuition behind is that information from other sensors might be correlated with the independent variable.

*4.2.3 Gradient Boosting Machines (GBM).* We use GBM as the third baseline method. GBM can extract non-linear patterns that LR and VAR cannot. The objective here is to minimize the error of the tree-based algorithm by adding weak learners with the help of a gradient descent like procedure. This algorithm follows a stepwise approach to minimize the loss and at every point, a new weak learner is added and the old ones are left unchanged.

## 4.3 Result and Discussion

In this section, we present the evaluation result of our model on forecasting $PM_{2.5}$ values for two cities, Los Angeles and Beijing. Table 2 shows the details about two datasets. We have observed that the Beijing dataset has a much larger standard deviation of $PM_{2.5}$ values than the Los Angeles dataset, so we expect that the air quality in Beijing is more challenging for the forecasting task. The $PM_{2.5}$ measurements are represented as $PM_{2.5}$ AQIs in the Los Angeles dataset and $PM_{2.5}$ concentrations in the Beijing dataset.

In graph construction, our model automatically selected important geographic feature types that have a great affect on $PM_{2.5}$ values. Table 3 shows the top 10 selected geographic feature types ranked by the importance in Los Angeles. Table 4 shows the top 10 important geographic feature types in Beijing. We have observed that "parking", "primary_road", "railways", and "motorway" are probably related to the traffic emission sources; "commercial", " hospital", "college" and "residential" are the places that attract people and traffic; "factory" could be the source of air pollutants; "pitch", "forest", and "park" are the open green areas, which might improve the air quality.

Table 5 shows the comparison of the forecasting results between the baseline methods (LR, VAR, and GBM) and the GC-DCRNN model in Los Angeles. Among the baselines, the VAR outperformed other baseline methods, which demonstrates that time series data from other locations can help improve the forecasting accuracy. Our GC-DCRNN model shows better performance than the baseline methods regarding both metrics when the horizon is larger than 6. However, when the horizon is small (h<=6), our model is slightly worse than the baseline methods, which might be due to the strong linearity when horizon is small (h<=6) in the Los Angeles dataset.

Table 6 shows that the GC-DCRNN model achieved the best performance among all the forecasting horizons with a 5%-10% improvement of MAE on the Beijing dataset. When the horizon grows, our model shows an significantly increasing improvement comparing with the baseline methods. This indicates our GC-DCRNN model can effectively exploit spatial dependency in forecasting large horizons and is more powerful in dealing with complex nonlinearity when the horizon increases. Compared with the result of modeling the Los Angeles dataset, we find that more sensors in the network graph might be helpful for the GC-DCRNN model since Beijing has more monitoring stations than Los Angeles does.

To investigate the effect of spatial dependency modeling, we designed two variants of the propose model, i.e., DCRNN with identity-graph (IG-DCRNN), which uses the identity matrix as the adjacency matrix, and geographical-distance based DCRNN (GD-DCRNN), which uses the geographical distance to measure the similarity and construct the adjacency matrix. Table 7 shows the comparison of GC-DCRNN, IG-DCRNN, and GD-DCRNN in term of MAE on the Beijing dataset. We have observed that the geo-context based DCRNN (GC-DCRNN) outperformed IG-DCRNN with an improvement in MAE from 2% to 9% and outperformed GD-DCRNN with an improvement in MAE from 1% to 5%, which shows the importance of appropriate spatial dependency modeling.

To further evaluate our model, we implemented a state-of-art hybrid method [34] and compared it with the GC-DCRNN model on the Beijing dataset. In [34], Zheng et al. separately built a temporal predictor (for dealing with the temporal relationship), a spatial predictor (for dealing with the spatial dimension), and then aggreated the results. In the experiment, Zheng et al. reported the forecasting results for the next six hours and the mean of max and min values in the next 7-12 and 13-24 hours against the mean of the real $PM_{2.5}$ values during the intervals. We reproduced their model (the temporal, spatial, and aggregation predictors) and generated the MAE results for the corresponding time intervals as mentioned in [34]. To compare with the hybrid model, we calculated the mean of 7-12 and 13-24 hours by aggregating the forecasting results from our model. Table 8 shows that our GC-DCRNN model outperformed the hybrid model regarding MAE at all forecasting horizons (h=1-6, 7-12, and 13-24), which demonstrates the advantage of simultaneously handling temporal and spatial dependencies in the GC-DCRNN model.

**Table 2: Details of Datasets**

| Datasets | Los Angeles | Beijing |
|---|---|---|
| Measurement Unit | AQI | Concentration |
| Time Span | 2017-01-01 -<br>2018-03-01 | 2017-01-01 -<br>2018-03-01 |
| Number of Stations | 9 | 35 |
| Number of Records | 132,288 | 355,950 |
| Missing Values (%) | 7.91 | 8.83 |
| Average | 50.72 | 53.76 |
| Standard Deviation | 26.95 | 61.60 |

**Table 3: Top 10 $PM_{2.5}$-related geographic feature types (ranked by the feature importance) in Los Angeles**

| Geo Name | Buffer Size (meter) | Geo type | Importance |
|---|---|---|---|
| land use | 1100 | parking | 0.07065 |
| land use | 2200 | pitch | 0.06250 |
| building | 1000 | commercial | 0.05000 |
| road | 3000 | primary_road | 0.04310 |
| building | 1900 | factory | 0.04310 |
| building | 700 | hospital | 0.04310 |
| building | 800 | college | 0.04310 |
| roads | 2000 | residential | 0.04153 |
| land use | 1000 | nature_reserve | 0.03750 |
| building | 2100 | factory | 0.03709 |

**Table 4: Top 10 $PM_{2.5}$-related geographic feature types (ranked by the feature importance) in Beijing**

| Geo Name | Buffer Size (meter) | Geo type | Importance |
|---|---|---|---|
| roads | 1200 | residential | 0.02133 |
| rail | 900 | railways | 0.01737 |
| land use | 2600 | park | 0.05000 |
| road | 300 | trunk | 0.01431 |
| land use | 1500 | forest | 0.01286 |
| roads | 2100 | service | 0.01282 |
| land use | 500 | park | 0.01245 |
| roads | 2000 | motorway_link | 0.01146 |
| land use | 1500 | retail | 0.01124 |
| roads | 2100 | primary_road | 0.01122 |

**Table 5: Comparison between the GC-DCRNN model and baseline models using $PM_{2.5}$ AQIs and meteorological data in Los Angeles**

| Horizon | Metric | LR | VAR | GBM | **DCRNN** |
|---|---|---|---|---|---|
| h=1 | MAE | 6.04 | 5.93 | 6.01 | **5.98** |
| | RMSE | 8.94 | 8.72 | 8.86 | **9.23** |
| h=6 | MAE | 14.33 | 13.81 | 13.83 | **13.86** |
| | RMSE | 18.7 | 18.01 | 17.87 | **18.43** |
| h=12 | MAE | 16.16 | 15.58 | 15.68 | **15.11** |
| | RMSE | 20.83 | 20.2 | 20.4 | **19.75** |
| h=18 | MAE | 17.06 | 16.83 | 16.99 | **15.58** |
| | RMSE | 21.95 | 21.56 | 21.91 | **20.48** |
| h=24 | MAE | 17.38 | 17.31 | 17.6 | **16.03** |
| | RMSE | 22.43 | 22.23 | 22.68 | **21.08** |

**Table 6: Comparison between the GC-DCRNN model and baseline models using $PM_{2.5}$ concentrations and meteorological data in Beijing**

| Horizon | Metric | LR | VAR | GBM | **DCRNN** |
|---|---|---|---|---|---|
| h=1 | MAE | 8.68 | 8.69 | 8.82 | **8.10** |
| | RMSE | 17.31 | 17.02 | 17.54 | **16.64** |
| h=6 | MAE | 25.52 | 22.69 | 25.4 | **22.05** |
| | RMSE | 41.05 | 37.89 | 41.36 | **37.85** |
| h=12 | MAE | 34.4 | 31.99 | 34.54 | **29.62** |
| | RMSE | 50.34 | 48.39 | 50.87 | **48.03** |
| h=18 | MAE | 39.02 | 38.06 | 39.51 | **34.38** |
| | RMSE | 53.27 | 53.32 | 54.25 | **52.05** |
| h=24 | MAE | 41.03 | 41.75 | 42.12 | **36.62** |
| | RMSE | 53.85 | 54.59 | 55.73 | **53.31** |

## 5 RELATED WORK

There exists an abundant literature working on real-time air quality forecasting (RT-AQF) for a short term or a long term depending on the applications [31, 32]. The short-term forecasts (1-5 days) are commonly used daily to inform the general public about the potential unhealthy air quality so that they can take preventive actions in advance. The long-term forecasts (>1 year) can provide variation trends of pollutants, which is often used by environmental

**Table 7: Effect of spatial dependency modeling. MAE results of the GC-DCRNN model and its variant, IG-DCRNN model and GD-DCRNN model.**

| Horizon | IG-DCRNN | GD-DCRNN | **GC-DCRNN** |
|---|---|---|---|
| h=1 | 8.94 | 8.13 | **8.10** |
| h=6 | 22.46 | 22.10 | **22.05** |
| h=12 | 31.24 | 30.14 | **29.62** |
| h=18 | 36.32 | 35.20 | **34.38** |
| h=24 | 39.72 | 37.45 | **36.62** |

**Table 8: MAE comparison between the hybrid model [34] and the GC-DCRNN model**

| Horizon | Hybrid Model [34] | **GC-DCRNN** |
|---|---|---|
| h=1 | 8.44 | **8.10** |
| h=2 | 13.07 | **12.15** |
| h=3 | 16.62 | **15.26** |
| h=4 | 19.45 | **17.85** |
| h=5 | 21.83 | **20.13** |
| h=6 | 23.86 | **22.05** |
| h=7-12 | 28.19 | **25.45** |
| h=13-24 | 34.41 | **31.43** |

health experts to analyze the climate change [29]. In [31, 32], short-term RT-AQF techniques are grouped into three categories: simple empirical approaches, statistical approaches, and physically-based approaches. Simple empirical approaches is usually not powerful enough to handle the air quality forecasting problem because it gives the result based on historical data that have similar conditions (e.g., temperature). Physically-based approaches usually require sound knowledge about the air pollutants as well as detailed data for analyzing the meteorological, physical, and chemical processes. However, the data are usually inaccessible to the public and the complex processes are difficult to be represented in a model. Therefore, statistical approaches or machine learning techniques become the most popular methods for the air quality forecasting problem [3, 13, 16, 34]. Auto-regressive integrated moving average (ARIMA) is a popular model for time series analysis and has been successfully applied to air quality forecasting [13, 14, 16]. ARIMA consists of three parts: 1) the auto-regressive (AR) part indicates that the evolving variable of interest can be approximated using a linear combination of its own historical values; 2) the moving average (MA) part models the residual from the AR part using a weighted combination of random noises at various previous time steps; 3) the integrate (I) part models the difference between adjacent values rather than the raw values. Other popular time series forecasting method [8] includes K-Nearest Neighbor (KNN), Support Vector Regression (SVR), Gaussian Process, etc. However, these time series models usually rely on the stationary assumption (i.e., the mean, variance, and autocorrelation structure do not change over time), which is often not suitable for real-time air quality data.

Artificial neural network (ANN) is also a popular method for air quality forecasting by modeling the non-linear temporal dependency. In [20], the author utilized multi-layer perceptron (MLP) artificial neural network (ANN) model to forecast daily maximum and average $O_3$ and particulate matter ($PM_{2.5}$ and $PM_{10}$) and showed that MLP outperformed the traditional multiple linear regression (MLR). However, it studied only one site "Chilliwack" with an extended period of air pollutant observations (three years). In [25], the authors proposed to build separate ANN models for each monitoring station to forecast the maximum value of the 24-hour moving average of $PM_{2.5}$. The experiment result showed that the multilayer neural network worked better than linear regression and the persistence baseline (i.e., assigning hourly values on the next day with the values at the present day). The approaches mentioned above show a relatively better performance of ANN on air quality forecasting, yet they deals with each time series separately without considering spatial dependency among them.

Some hybrid models were proposed to handle both linear and non-linear patterns in air quality time series data. In [7], the authors presented a hybrid model combining ARIMA and ANN to improve the forecasting accuracy for an area with limited air quality and meteorological data. The idea was to first build an ARIMA model to forecast daily maximum $PM_{10}$ value and then use an ANN model to describe the residuals from the ARIMA model. The result reported that the hybrid model outperformed the separate ARIMA model and ANN model. Similarly, in [3], the authors proposed a hybrid model by combining an ARIMA model and a non-linear model. The experiment showed an improvement on the hybrid model comparing to the separate ARIMA model and the non-linear model with a reduction of 26.31% and 21.05% in terms of the relative error (the ratio of the absolute error to the real measurements). However, those hybrid approaches do not consider spatial correlation with the air quality data from other monitoring stations. In [34], the authors proposed a hybrid model to forecast the air quality over the next 48 hours (i.e., real-valued AQIs for the next 6 hours and a max-min range of AQI for the next 7-12, 12-24 and 24-48 hours) for each monitoring station. The hybrid model could handle the temporal and spatial dependency in separate models (i.e., temporal predictor and spatial predictor) and aggregate two predictors with a regression tree. The result showed that the hybrid model outperformed the individual models. Some fixed-size buffers were selected to pick the neighborhood stations around the target stations. However, geographic distances cannot reflect the similarity between the air quality readings of two sensors when they are far apart from each other. Moreover, separate models might not represent the spatiotemporal patterns as a whole comprehensively.

Deep learning approaches deliver new promise for the time series forecasting problem. Deep recurrent neural network (RNN), which is able to model non-linear temporal dependency, has recently achieved promising results in sequence modeling as well time series modeling [17], e.g., speech recognition [28] and traffic forecasting [30]. To jointly model the spatial dependency and the temporal dependency among time series, Li et al. [18] proposed the Diffusion Convolutional Recurrent Neural Network (DCRNN) which combines diffusion convolution with RNN.

Our proposed model, the GC-DCRNN model, is distinguished from the above approaches in a number of ways: 1) our model manipulates the spatial dependency with regard to air quality by automatically selecting the crucial geographic features in neighbors that have a great impact on $PM_{2.5}$ concentrations and constructing the graph with the similarity of the important features between sensors instead of geographic distance; 2) our model captures both the spatial and temporal dependencies in one model instead of just handling temporal dependency or dealing with them in separate models. The model can generate accurate air quality forecasting results in the following 24 hours for all the stations at one time.

## 6 CONCLUSIONS AND FUTURE WORK

This paper presented a data driven approach to forecast $PM_{2.5}$ concentrations using the previous-hour air quality and meteorological data. The advantages of our approach include 1) our model could handle both spatial and temporal dependencies in the time series data simultaneously and achieved a better performance than other traditional methods; 2) our method represents the spatial correlation in a graph with automatically selected important geographic feature types that largely affect $PM_{2.5}$ concentrations and uses those important geographic feature types to compute the adjacency graph for the DCRNN model; 3) we use the easily accessible OpenStreetMap to construct the geographic abstraction for capturing the spatial dependency among air quality data instead of using data that is expensive and difficult to obtain (e.g., traffic data). We plan to include other important air quality-related features for handling other temporal dynamics, e.g., workday/weekend and seasonal effects. Also, we plan to test more datasets, such as PurpleAir data, and other geogeographic regions.

## ACKNOWLEDGMENTS

This work is supported in part by the NIH grant 1U24EB021996-01, LA Metro contract LA-Safe-PS36665000, Caltrans-65A0533, and NVIDIA Corporation.

## REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*, Vol. 16. 265–283.

[2] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. 2015. Time-series clustering–A decade review. *Information Systems* 53 (2015), 16–38.

[3] Asha B. Chelani and Sukumar Devotta. 2006. Air quality forecasting using a hybrid autoregressive and nonlinear model. *Atmospheric Environment* 40, 10 (2006), 1774–1780.

[4] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* abs/1412.3555 (2014).

[5] W. Geoffrey Cobourn. 2007. Accuracy and reliability of an automated air quality forecast system for ozone in seven Kentucky metropolitan areas. *Atmospheric Environment* 41, 28 (2007), 5863–5875.

[6] W. Geoffrey Cobourn and Milton C. Hubbard. 1999. An enhanced ozone forecasting model using air mass trajectory analysis. *Atmospheric Environment* 33, 28 (1999), 4663–4674.

[7] Luis A. Díaz-Robles, Juan C. Ortega, Joshua S. Fu, Gregory D. Reed, Judith C. Chow, John G. Watson, and Juan A. Moncada-Herrera. 2008. A hybrid ARIMA and artificial neural networks model to forecast particulate matter in urban areas: The case of Temuco, Chile. *Atmospheric Environment* 42, 35 (2008), 8331–8340.

[8] James Douglas Hamilton. 1994. *Time series analysis*. Vol. 2. Princeton university press Princeton.

[9] K.-I. Hoi, Ka-Veng. Yuen, and Kai-Meng Mok. 2008. Kalman filter based prediction system for wintertime PM10 concentrations in Macau. *Global NEST Journal* 10, 2 (2008), 140–150.

[10] Héctor Jorquera, Ricardo Pérez, Aldo Cipriano, Andrés Espejo, M. Victoria Letelier, and Gonzalo Acuña. 1998. Forecasting ozone daily maximum levels at Santiago, Chile. *Atmospheric Environment* 32, 20 (1998), 3415–3424.

[11] Marilena Kampa and Elias Castanas. 2008. Human health effects of air pollution. *Environmental pollution* 151, 2 (2008), 362–367.

[12] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems* 3, 3 (2001), 263–286.

[13] Anikender Kumar and P. Goyal. 2011. Forecasting of daily air quality index in Delhi. *Science of the Total Environment* 409, 24 (2011), 5517–5523.

[14] Ujjwal Kumar and V.K. Jain. 2010. ARIMA forecasting of ambient air pollutants (O3, NO, NO2 and CO). *Stochastic Environmental Research and Risk Assessment* 24, 5 (2010), 751–760.

[15] Nino Künzli, Michael Jerrett, Wendy J. Mack, Bernardo Beckerman, Laurie LaBree, Frank Gilliland, Duncan Thomas, John Peters, and Howard N Hodis. 2005. Ambient air pollution and atherosclerosis in Los Angeles. *Environmental health perspectives* 113, 2 (2005), 201.

[16] Muhammad Hisyam Lee, Nur Haizum Abd Rahman, Mohd Talib Latif, Maria Elena Nor, and Nur Arina Bazilah Kamisan. 2012. Seasonal ARIMA for forecasting air pollution index: A case study. *American Journal of Applied Sciences* 9, 4 (2012), 570–578.

[17] Yaguang Li and Cyrus Shahabi. 2018. A brief overview of machine learning methods for short-term traffic forecasting and future directions. *SIGSPATIAL Special* 10, 1 (2018), 3–9.

[18] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

[19] Yijun Lin, Yao-Yi Chiang, Fan Pan, Dimitrios Stripelis, José Luis Ambite, Sandrah P Eckel, and Rima Habre. 2017. Mining public datasets for modeling intra-city PM2. 5 concentrations at a fine spatial resolution. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 25.

[20] Ian G. McKendry. 2002. Evaluation of artificial neural networks for fine particulate pollution (PM10 and PM2.5) forecasting. *Journal of the Air & Waste Management Association* 52, 9 (2002), 1096–1101.

[21] Clare S Murray, Gina Poletti, Tatiana Kebadze, Julie Morris, Ashley Woodcock, S. L. Johnston, and Adnan Custovic. 2006. Study of modifiable risk factors for asthma exacerbations: virus infection and allergen exposure increase the risk of asthma hospital admissions in children. *Thorax* 61, 5 (2006), 376–382.

[22] World Health Organization and UNAIDS. 2006. *Air quality guidelines: global update 2005*. World Health Organization.

[23] E. Patterson and D. J. Eatough. 2000. Indoor/outdoor relationships for ambient PM2.5 and associated pollutants: epidemiological implications in Lindon, Utah. *J. Air Waste Manag. Assoc.* 50, 1 (2000), 103–110.

[24] Patricio Perez and Giovanni Salini. 2008. PM2.5 forecasting in a large city: comparison of three methods. *Atmospheric Environment* 42, 35 (2008), 8219–8224.

[25] Patricio Pérez, Alex Trier, and Jorge Reyes. 2000. Prediction of PM2.5 concentrations several hours in advance using neural networks in Santiago, Chile. *Atmospheric Environment* 34, 8 (2000), 1189–1196.

[26] Victor R Prybutok, Junsub Yi, and David Mitchell. 2000. Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations. *European Journal of Operational Research* 122, 1 (2000), 31–40.

[27] David Y.H. Pui, Sheng-Chieh Chen, and Zhili Zuo. 2014. PM2. 5 in China: Measurements, sources, visibility and health effects, and mitigation. *Particuology* 13 (2014), 1–26.

[28] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger (Eds.). Curran Associates, Inc., 3104–3112.

[29] Amos PK Tai, Loretta J Mickley, and Daniel J Jacob. 2010. Correlations between fine particulate matter (PM2.5) and meteorological variables in the United States: Implications for the sensitivity of PM2.5 to climate change. *Atmospheric Environment* 44, 32 (2010), 3976–3984.

[30] Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. 2017. Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting. In *SIAM International Conference on Data Mining*. 777–785.

[31] Yang Zhang, Marc Bocquet, Vivien Mallet, Christian Seigneur, and Alexander Baklanov. 2012. Real-time air quality forecasting, part I: History, techniques, and current status. *Atmospheric Environment* 60 (2012), 632–655.

[32] Yang Zhang, Marc Bocquet, Vivien Mallet, Christian Seigneur, and Alexander Baklanov. 2012. Real-time air quality forecasting, part II: State of the science, current research needs, and future prospects. *Atmospheric Environment* 60 (2012), 656–676.

[33] Mei Zheng, Lynn G. Salmon, James J. Schauer, Limin Zeng, C. S. Kiang, Yuanhang Zhang, and Glen R. Cass. 2005. Seasonal trends in PM2. 5 source contributions in Beijing, China. *Atmospheric Environment* 39, 22 (2005), 3967–3976.

[34] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. 2015. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2267–2276.