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

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

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

ABSTRACT

Forecasting spatial correlated time series 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., autoregressive integrated moving average (ARIMA) and Artificial Neural Network (ANN), either fails to model the non-linear temporal dependency or cannot appropriately consider spatial relationships between multiple spatial time series data. In this paper, we present an approach for forecasting the short-term $PM_{2.5}$ concentrations, using a deep learning model, diffusion convolutional recurrent neural network (DCRNN). The model describes the spatial relationship by constructing a graph based on the similarity between sensor locations computed using their surrounding "important" geographic features regarding their impacts to air quality (e.g., the area size of parks within a 1000-meter buffer, the number of factories within a 500-meter buffer). Besides, 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 μm. Typically, vehicle emissions, industrial sources, and burning sources (e.g., wildfires and coal combustion) are the primary contributors to generate fine particulate matter. Many epidemiological studies [21] [25] 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 raise 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. Besides, fine particulate matter and its derivatives also cause many adverse effects on the environment such as poor visibility, global climate change [27], and ecological damage.

In 2006, the World Health Organization (WHO) suggested that long-term exposures, i.e., an annual average of $PM_{2.5}$ concentrations, should not exceed 10mg/m^3, and the 24-h average of $PM_{2.5}$ concentrations should not exceed 25mg/m^3 [20]. Many countries also 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 matter ($PM_{2.5}$ and PM_{10}). In the United States, the Environmental Protection Agency (US EPA) establishes ambient air monitoring networks, which provide hourly measurements for various pollutants (e.g., $PM_{2.5}$ and PM_{10}). US EPA also sets a national air quality standard, the air quality index (AQI), to indicate health risk level of current pollution level. US EPA calculates the AQI values for five major air pollutants[1] : 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 μg/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 definition for each category. Generally, an AQI value below 100 is an acceptable range. While the value is rising above 100, it is considered to be unhealthy for certain sensitive groups, and then the public should take protective actions as the value is getting higher.

ACM SIGSPATIAL, November 2018, Seattle, Washington, USA

ACM ISBN 123-4567-24y-567/08/06…$15.00
https://doi.org/10.475/123_4

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

Besides establishing monitoring stations to report the real-time air quality status, there is an increasing demand to forecast the air quality pollutants in the future, which not only supports governments to make policies like pollution control but also informs the general public to take advanced actions like staying at home or outside for hiking. However, forecasting air quality is a challenging task. First, air quality values can vary significantly over time and across locations. Figure 1 shows the $PM_{2.5}$ AQI values at two reporting areas in Los Angeles, "Central LA CO" and "Santa Clarita Vly" during the period in early January and July. We observe that "Central LA CO" generally shows higher $PM_{2.5}$ AQI values than that in "Santa Clarita Vly" (might due to heavier traffic congestion in "Central LA CO"). Besides, the $PM_{2.5}$ AQI values are significantly higher in July than in January (might due to holiday events in July). Second, sudden changes in the observations make it more 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 factory emissions. Figure 1 shows extremely high $PM_{2.5}$ AQI values during the Independence Day (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 chemical processes of air pollutants. This challenge together with the fact that air quality monitoring stations are usually sparse, make it difficult to use geographic distance to capture the spatial relationships between air quality data [18].

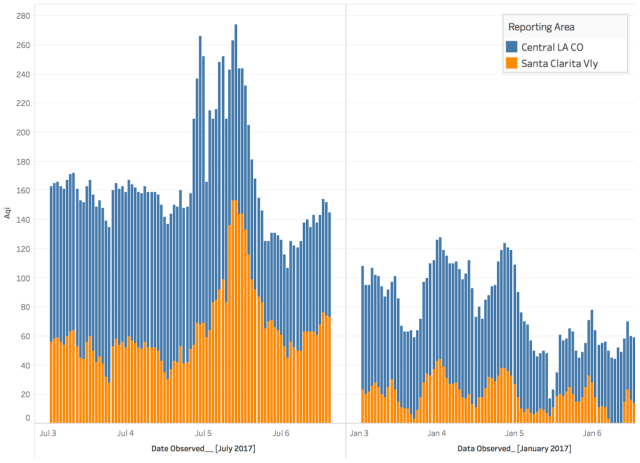

Figure 1: An example of $PM_{2.5}$ AQIs at two Los Angeles reporting areas in January and July

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, 24] for computing future air quality values 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 [23]. ARIMA and KF only work for stationary time series data failing to capture the dynamics and trends in the air quality data, while Regression methods like Linear Regression are incapable of handling the non-linearity in time series data. Artificial Neural Network (ANN) based methods[7, 19, 22, 23] have also 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 [19, 22].

A general way to define the spatial dependency is using geographic distance [31], i.e., if a sensor location is close to the target location, the sensor is considered as the neighbor that will contribute to forecasting the air quality at the target location. However, the geographic distance would not work if the monitoring stations are sparse and hence geographic distances cannot fully capture the similarity in environmental characteristics (like near industrial area or green land), which have various impacts on air quality.

In [31], the authors proposed a hybrid model to separately deal with temporal and spatial correlations. The hybrid 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 previous two predictors, and an inflection predictor to detect 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. Also, it uses separate models for temporal and spatial patterns correlation, which might not comprehensively represent the spatiotemporal patterns as an entirety. It remains challenging 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 deep learning model to forecast the $PM_{2.5}$ concentrations in the next several hours (e.g., 6, 12, 24 and 48) at a given location. In our method, we utilize the Diffusion Convolutional Recurrent Neural Network (DCRNN) [17] to build air quality forecasting model, which has achieved state-of-the-art performance in traffic forecasting. The DCRNN model is able to manipulate spatial and temporal dependency together and also we use the neighborhood characteristics to represent spatial correlation in a graph, which means two locations are similar if they share a similar built environment. The following datasets are employed to construct the model: 1) the $PM_{2.5}$ concentrations time series data, which is for training and validating the model; 2) the meteorological time series data, serving as auxiliary features in the model; 3) the geographic data that works for building a graph that denotes the spatial relationship between monitoring stations. We define the spatial correlation as the "similarity" or "distance" between two stations. Traditional methods compute the geographic distance as proximity based on the idea that the closer two locations, the higher the spatial correlation (i.e., they will have similar air quality value). However, when the stations are far apart from each other, the geographic distance becomes less informative as all distances are similarly high values in this situation. Therefore, We employ our previous work on air quality prediction [18] to automatically select the "important" geographic feature types (e.g., factories within

1,000 meters) that have a significant impact on air quality for a given location. With the "important" geographic features, we create a graph in which the nodes are sensors and edge weights are the similarity between geographic features around sensors. Then, the DCRNN model utilizes the diffusion convolution on this pre-constructed 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 convolution operation into the recurrent neural network, and The inputs of the model is a sequence of air quality readings and related meteorological data (e.g., humidity, temperature, and wind speed) over past few hours at all the stations. While the outputs are the forecasting $PM_{2.5}$ concentrations, i.e., a sequence of values for next 24 hours (section 3), at all the stations. In our experiment, DCRNN shows a consistently better performance than other air quality forecasting baselines (section 4). Our main contribution is that we utilize the Diffusion Convolutional Recurrent Neural Network (DCRNN) to jointly manipulate spatial and temporal dependencies in location-dependent air quality time-series data. We use an automatic approach to describe spatial dependency by considering the similarity in the surrounding built environment. We construct a graph enabling the DCRNN to handle the spatial correlations in air quality data by automatically selecting important geographic feature types that have a significant impact on air quality pollutants. We conduct extensive experiments on two real-world datasets collected in Los Angeles and Beijing, and observe clear improvements of 5%-10% over baseline approaches.

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 about 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

2.1 AQS (Air Quality System) Data

2.1.1 Beijing air quality data from Biendata. Biendata provides publicly accessible air quality data of Beijing in the KDD CUP of Fresh Air[2]. The data have hourly air quality concentrations for various pollutants including $PM_{2.5}$, PM_10, NO_2, CO, O_3, and SO_2. As figure 2 shows, there is a total of 35 monitoring stations in Beijing. In this paper, we use the $PM_{2.5}$ concentrations from 2017-01-01 00:00:00 to 2018-03-01 00:00:00 for both Beijing and Los Angeles for training and testing our air quality modeling approach. Note that the approximate area size is 6,490 miÂš for Beijing and 4,751 miÂš for Los Angeles, and the Beijing datasets include nearly four times more reporting stations than the Los Angeles datasets (35 vs. 9).

2.1.2 Los Angeles air quality data from EPA. We are collecting the air quality data, including $PM_{2.5}$, 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 (see Figure 3). In the experiment, we remove 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 duplicates with other areas. Therefore, our model utilizes the $PM_{2.5}$ AQI observations from 9 reporting areas with the period from 2017-01-01 00:00:00 to 2018-03-01 00:00:00 with a one-hour interval. Table 1 gives an example of the structured $PM_{2.5}$ AQI data.

Table 1: Examples 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

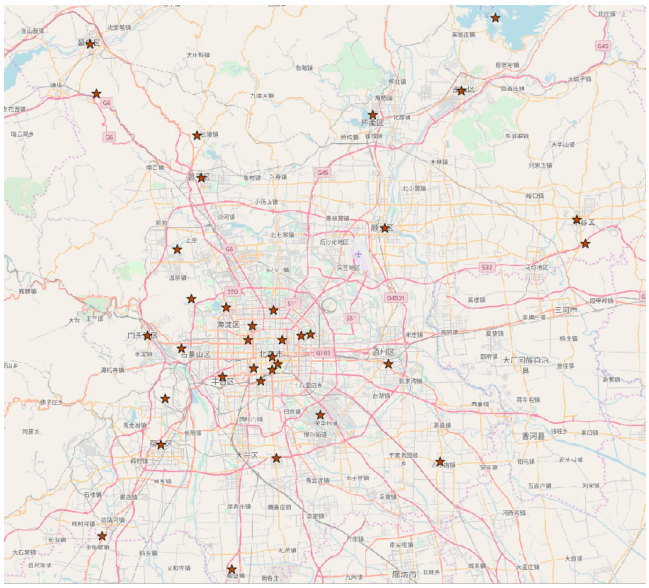

Figure 2: Monitoring Station Locations in Beijing

2.2 Meteorological Data

We collect meteorological data through the Dark Sky API. [4] Dark Sky reports fine-scale weather data (temperature, humidity, wind speed, wind direction, etc.) all over the world. For each reporting area or monitoring station in Beijing and Los Angeles, we query the meteorological data based on the coordinates of the monitoring station locations every hour. The meteorological data have the same time period (i.e., from 2017-01-01 00:00:00 to 2018-03-01 00:00:00) and time resolutions (i.e., hourly) as the air quality data.

[2] https://biendata.com/competition/kdd_2018/data/
[3] https://docs.airnowapi.org/webservices
[4] https://darksky.net/dev/docs

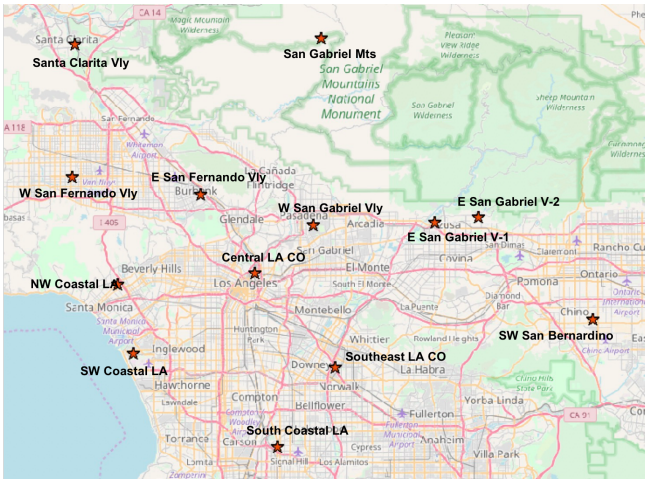

Figure 3: Reporting Area Locations in Los Angeles

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 [5]. For example, roads, represented as lines, contain many types such as motorway and pedestrian roads. Land uses describe the function of an areas, such as industrial area and commercial area. Water areas represent the areas of lakes or ponds. In this paper, we utilize OpenStreetMap data to construct the geographic data.

3 METHODOLOGY

The goal is to forecast the future $PM_{2.5}$ concentrations in next 24 hours at a given location. We use 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 features on each node (e.g., the air quality reading and meteorological features). Let $\boldsymbol{X}^{(t)}$ represent 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:

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

3.1 Graph Construction

In this section, we briefly describe our previous work that utilizes the similarity of the "important" geographic features around monitoring stations to represent their spatial dependency and construct the graph for diffusion convolution later [18]. Geographic distances become less informative when the sensors are far from each other. Therefore, we take advantage of our previous work [18], which uses a data-driven method to automatically select critical geographic features that have a significant impact on air quality.

3.1.1 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 in mountain areas, so they should be grouped into separate clusters. [2] shows that traditional cluster methods like K-means do not work for high dimensional time series data. Due to the high dimension of our air quality data (around 7,000), we use the piece-wise aggregate approximation (PAA) [12] to reduce the dimension by representing original sequences with the daily average of highest three values and lowest three values. Then we use K-means to group the lower dimensional data after applying PAA with K at elbow point to get the clustering result.

3.1.2 Constructing Geographic Abstraction. Our approach generates ambient geographic features for each monitoring station using OpenStreetMap data. Figure 4 is an example of how we construct the geographic abstraction. There are various geographic features, including roads, land uses, and buildings, around the monitoring station A with the 100-meter (m) and 200m 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 $500m^2$ water area within 100 meter (m) buffer while $950m^2$ water area and $740m^2$ green land within 200m buffer. Thus, it generates the geographic abstraction vector as [500, 0, 950, 740]. For the line features, like roads and aeroways, our approach computes the sum of lengths of various feature types as geographic abstraction. Monitoring station A has 23m pedestrian and 30m motorway within 100m buffer while 43m Pedestrian and 200m Motorway within 200m buffer. Thus, it generates the geographic abstraction vector as [23, 30, 43, 200]. For point features, like buildings, our approach counts the number of various feature types. Monitoring station A has 2 houses and 2 commercial buildings within 100m buffer while 6 houses and 5 commercial buildings within 200m buffer. Thus, it generates the geographic abstraction vector as [2, 2, 6, 5]. Our approach generates whole vector as geographic abstraction for each location, like [500, 0, 950, 740, 23, 30, 43, 200, 2, 2, 6, 5] for the example. In practice, we create buffers from 100m to 3,000m with an interval of 100m and generate the value for each unique geographic feature type with the buffers.

3.1.3 Computing Geo-Context. Our approach utilizes the clustering result and the geographic abstraction to automatically identify the important geographic feature types with what buffer size that has the most impact on $PM_{2.5}$ concentrations. A random forest classifier is used to quantify the importance of individual components in the geographic abstraction vector. For unimportant features, the model gives the importance as zero. We keep those important ones (non-zeroes) to form a new vector, called geo-context. To construct the graph, we compute the similarity between the geo-context of the monitoring stations as the edge weight by using Euclidean distance. In the next step, we would embed the graph in the DCRNN model to handle the spatial dependency.

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

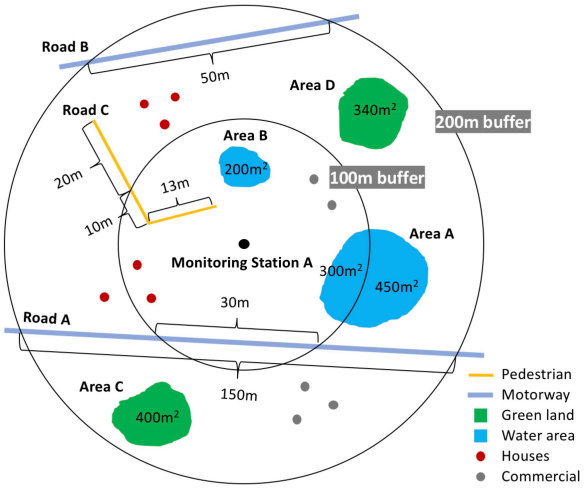

Figure 4: Examples of geographic features in the 100-meter and 200-meter buffers

3.2 Diffusion Convolution

In this section, we propose diffusion convolution 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 3×3 filter is defined to add all the elements within the filter, a 5×5 input image maps to a 3×3 output image whose elements are the sum of each 3×3 grids based on the filter. The diffusion convolution extends this idea to the general graph-structured data. In DCRNN [17], it defines diffusion convolution as the combination of diffusion processes with different steps on the graph. Specifically, it defines the K diffusion steps, which represent the "distance" to the center point (i.e., the current forecasting location). As the figure 6 shows, at each step k, it looks at the neighbors that k-step away from the center point and computes the transition matrices for this step. The diffusion convolutional filter then adds the transition matrices with some probability θ, 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_θ is defined as:

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

where $\boldsymbol{\theta} \in \mathbb{R}^{K \times 2}$ are the parameters for the filter, D denotes the diagonal degree matrix 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, a diffusion convolutional layer is build to map P-dimensional features to Q-dimensional output, which is similar to extracting features in CNN.

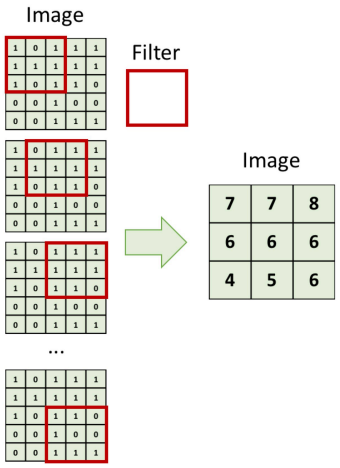

Figure 5: An example of Convolutional Neural Networks

3.3 Deep Learning Model - DCRNN

Besides spatial dependency, the model also 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 current input as traditional machine learning algorithms do, but also it makes use of the information from previous cells. 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 previous hidden cell $\boldsymbol{H}_{t-1}$ and current input 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 gated recurrent units (GRU) [4] as the cell, which is a simple yet powerful variant of RNN. As Figure 9 shows, GRU contains two gates: 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. 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. $\boldsymbol{b}_c$ is the bias. $\odot$ is the Hadamard product, which multiply the elements on corresponding location in the two matrix.

[6]Reprinted from the poster of [17] with permission from the corresponding author.

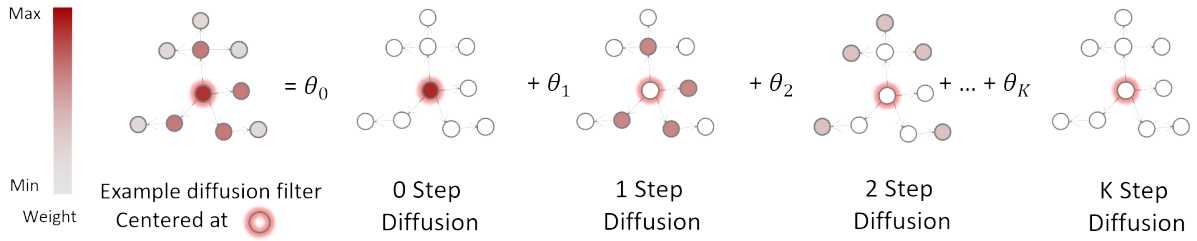

Figure 6: Example Diffusion Process [6]

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

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

In order to capture the spatio-temporal dependencies, DCRNN replaces the matrix multiplication in GRU with the diffusion convolution, i.e., diffusion convolutional gated recurrent unit (DCGRU).

$$r^{(t)} = \sigma(\Theta_r \star_G [X^{(t)}, H^{(t-1)}] + b_r)$$
$$u^{(t)} = \sigma(\Theta_u \star_G [X^{(t)}, H^{(t-1)}] + b_u)$$
$$C^{(t)} = \tanh(\Theta_C \star_G [X^{(t)}, (r^{(t)} \odot H^{(t-1)})] + b_c)$$
$$H^{(t)} = u^{(t)} \odot H^{(t-1)} + (1 - u^{(t)}) \odot C^{(t)}$$

where $\star_G$ denotes the *diffusion convolution* defined in Equation 1 and Θ_r,Θ_u,Θ_C are parameters for the corresponding filters.

To conduct the multi-step ahead forecasting, DCRNN utilizes the *Sequence to Sequence* architecture [26]. 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. Here both the encoder and decoder are recurrent neural network with DCGRU. In this way, it generates air quality forecasting results given previous hours data by handling both spatial and temporal dependency simultaneously.

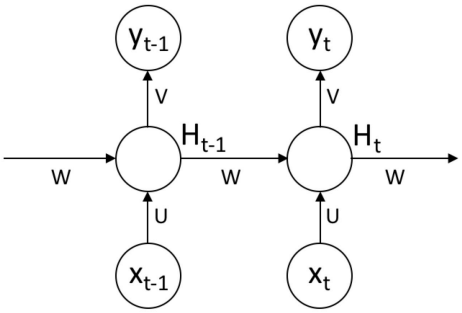

Figure 7: An example of Recurrent Neural Networks

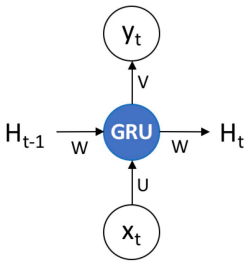

Figure 8: An example of Gated Recurrent Units

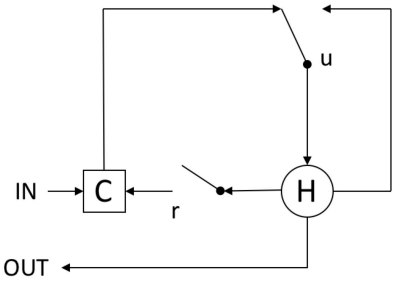

Figure 9: Gated Recurrent Units Structure

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 described in Section 2. We conducted the experiment in a Docker container deployed on a Linux workstation with 4 physical cores and 64GB memory. All geospatial computing was done in PostGIS, the baseline models were implemented with Python 2.7 and with the Tensorflow [1] framework.

4.1 Environmental settings

In the experiment, we tested the performance of our air quality forecasting model. We split the air quality data into training data (from 2017-01-01 to 2017-12-31) and testing data (from 2018-01-1 to 2018-03-01). Our approach uses previous 24-hour data (i.e., sequence

length = 24) to forecast next 24 hours $PM_{2.5}$ concentrations (i.e., horizon = 24). 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 observed samples. In the experiment, we evaluate the model by comparing our results with baseline methods by using the following metrics and missing values are 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, as the formula defined:

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

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

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

3. Mean Absolute Percentage Error (MAPE) evaluates regression models as it expresses the error as a percentage with respect to the ground truth and is given by the formula:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} |\frac{y_i - \hat{y}_i}{y_i}|$$

4.2 Baseline methods

4.2.1 Linear Regression. Linear Regression (LR) can capture linear dependencies between the input variables (e.g., features) and the output variables (e.g., predictions). These predictions are autocorrelated, 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, it captures the autocorrelation of variables and with some lag variables emulates Autoregression, a time series analysis technique. For example, we 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. Vector Autoregression Regression (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 the sensor, all other time series from other sensors are fed into the model. The intuition behind this technique is that information from other sensors might be correlated with the independent variable.

4.2.3 Gradient Boosting Machines. We use Gradient Boosting Machines (GBM) as the third baseline method. GBM can extract nonlinear patterns that LR and VAR are not capable of. 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 step-wise approach to minimize the loss and at every point, a new weak learner is added and the old ones are left unchanged.

4.2.4 Random Forest. Random Forest (RF) is a state-of-the-art tree-based learning algorithm that works on the principle of bootstrapping. The objective is to create multiple trees with a subset of the features chosen randomly at each node and take into account the average prediction of all trees in the regression task. The process allows decorrelation of trees and also could alleviate the overfitting problem generally observed in the decision tree algorithm, where the trees are constructed very deep [1]. The idea behind using this technique is the same as GBMs, i.e., capturing non-linearity in the data.

4.3 Data Preprocessing

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

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 is untouched. In the experiment, we utilize whole year data in 2017 as training data, 10% of the training data as validation data, and two months in 2018 as testing data.

4.4 Result and Discussion

In this section, we present the evaluation 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 observe Beijing dataset has a much larger standard deviation than Los Angeles dataset, so it would be more challenging for the air quality forecasting task. The $PM_{2.5}$ measurements are represented as $PM_{2.5}$ AQIs in Los Angeles dataset. In Beijing dataset, the measurement unit is $PM_{2.5}$ concentrations. In graph construction, our model automatically selects important geographic feature types that effect $PM_{2.5}$ values. Table 3 shows the top 10 important geographic feature types in Beijing. Table 4 shows the top 10 selected geographic feature types ranked by importance in Los Angeles. We observe 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" is the source of air pollutants; "pitch", "forest", and "park" are the open green areas, which might have a positive effect on air quality. In this way, we could easily explain what geographic feature types affect $PM_{2.5}$ concentrations and from what distance. Then we compute the similarity based based on these air-quality-related features to construct the adjacency graph for DCRNN.

The input to the DCRNN model is the constructed graph, $PM_{2.5}$ measurements and meteorological data (temperature, humidity, wind speed, wind direction, etc.). We set the parameters *sequence length* equals to 24 and *horizon* is 24, which means the model will utilize previous 24-hour $PM_{2.5}$ and meteorological readings to form

Table 2: Details of Datasets

Datasets	Los Angeles	Beijing
Time Span	2017-01-01 - 2018-03-01	2017-01-01 - 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

a sequence as "x" to forecast next 24-hour $PM_{2.5}$ values (as "y") sequentially. We report the result for the temporal horizons equal to 6, 12, 18 and 24 hours with three metrics, MAE, RMSE, and MAPE.

Table 5 shows that our model achieved the best performance regarding all the metrics for all forecasting horizons with a 5%-10% improvement in Beijing dataset. When the horizon is growing, our model shows an increasing improvement on all metrics outperforming the baseline methods especially for MAPE. This indicates our context-based DCRNN can effectively exploiting spatial dependency in forecasting large horizon and also our model is more powerful in dealing with complex non-linearity when the horizon increases. We also implement an aggregation method for Beijing's result as Zheng et al. [31] did in the model evaluation. We compute the mean of max and min value in the next 7-12 and 13-24 hours against the mean of the real $PM_{2.5}$ values during the intervals. The mean of our dataset is 53.76, which is lower than their dataset (106.4). We achieve an MAE value=25.45759 for 7-12h and 31.43901 for 13-24h while in [31], they claim MAE value=52.4 for 7-12h and 63.9 for 13-24h in Beijing.

Table 6 shows the results for the baseline methods and the DCRNN model in Los Angeles. Among the baseline methods, VAR outperforms others, which demonstrates time series data from other locations would help improve the forecasting accuracy. We observe that when the horizon equals to 6, the MAE and RMSE of our method fail to exceed baselines, but MAPE is lower than the baselines. The reason could be that our model performed when the ground truth is small (e.g., low AQI readings). For example, suppose the ground truth at two timestamps are t_1=100 and t_2=10, our forecasting results are t_1=90 and t_2=9 and the baseline results are t_1=95 and t_2=15. In this case, the MAE and RMSE of our model is higher while MAPE is relatively lower than the baseline. When the horizon is growing, our model shows a better results in all metrics. We performed the paired t-test and find that the DCRNN MAE results and VAR, which has the best performance among baselines, are statistically different (for horizon 6, 12, 18, 24 are p=5.45E-4, p=7.40E-07, p=2.22E-13, p=8.13E-18, respectively).

Since Beijing has more monitoring stations than Los Angeles, we find that more sensors in the network graph might be helpful for DCRNN model. To evaluate the effect of spatial modeling, we design a variant of the propose model, i.e., DCRNN-IG, which uses the identity matrix as the adjacency matrix. Table 7 shows the comparison of DCRNN-IG and DCRNN, which uses the graph constructed with geo-context, in term of MAE on the Beijing dataset. We observe that DCRNN consistent outperform its variant which justify the importance of appropriate spatial dependency modeling.

Table 3: Top 10 geographic feature types (ranked by 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 4: Top 10 geographic feature types (ranked by 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 5: Forecasting results for baseline models and DCRNN model using $PM_{2.5}$ concentrations and meteorological data in Beijing

Horizon	Metric	LR	VAR	GBM	**DCRNN**
h=6	MAE	25.52	22.69	25.4	**21.43**
	RMSE	41.05	36.89	41.36	**37.85**
	MAPE	170.00%	144.00%	169.00%	**108.94%**
h=12	MAE	34.4	31.99	34.54	**29.70**
	RMSE	50.34	47.39	50.87	**48.03**
	MAPE	253.00%	238.00%	257.00%	**180.25%**
h=18	MAE	39.02	38.06	39.51	**34.83**
	RMSE	53.27	52.32	54.25	**52.17**
	MAPE	291.00%	285.00%	297.00%	**220.39%**
h=24	MAE	41.03	41.75	42.12	**37.62**
	RMSE	53.85	54.59	55.73	**53.46**
	MAPE	305.00%	307.00%	318.00%	**235.81%**

5 RELATED WORK

There exists an abundant literature working on real-time air quality forecasting (RT-AQF) for short-term or long-term depending on the application objectives [29, 30]. 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

Table 6: Forecasting results for baseline models and DCRNN model using $PM_{2.5}$ AQIs and meteorological data in Los Angeles

Horizon	Metric	LR	VAR	GBM	RF	**DCRNN**
h=6	MAE	14.33	13.81	13.63	13.86	**14.00**
	RMSE	18.7	18.01	17.87	18.1	**19.29**
	MAPE	60.00%	58.00%	59.00%	59.00%	**49.32%**
h=12	MAE	16.16	15.58	15.68	15.92	**15.32**
	RMSE	20.83	20.2	20.4	20.66	**21.01**
	MAPE	71.00%	69.00%	70.00%	69.00%	**56.24%**
h=18	MAE	17.06	16.83	16.99	17.27	**15.65**
	RMSE	21.95	21.56	21.91	22.27	**21.31**
	MAPE	77.00%	78.00%	78.00%	77.00%	**59.14%**
h=24	MAE	17.38	17.31	17.6	17.82	**15.92**
	RMSE	22.43	22.23	22.68	23.01	**21.62**
	MAPE	79.00%	81.00%	81.00%	79.00%	**60.64%**

Table 7: Effect of spatial dependency modeling. MAE comparison of DCRNN and its variant DCRNN-IG which uses the identity matrix as the graph.

Horizon	DCRNN-IG	**DCRNN**
h=6	22.46	**21.43**
h=12	31.24	**29.70**
h=18	36.32	**34.83**
h=24	39.72	**37.62**

variation trends of pollutants, which is often used by environmental health experts to analyze global climate change. In [29, 30], 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 air pollutants as well as detailed data for analyzing meteorological, physical, and chemical processes. However, the data is usually inaccessible to the public and the complex processes are hard to be represented in a model. Therefore, statistical approaches or machine learning techniques are the most popular methods in recent work. 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 raw values. Other popular time series forecasting method [8] includes K-nearest Neighbor (KNN), Support Vector Regression (SVR), particle filter, Gaussian Process etc. However, these time series models usually rely on the stationary assumption, 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. [19] utilizes 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 shows that MLP is superior to traditional multiple linear regression (MLR). However, it studied only one site "Chilliwack" with an extended period of air pollutant observations (3 years). In [23], it builds separate ANN models for each monitoring station to forecast the maximum value of the 24-h moving average of $PM_{2.5}$. The experiment result shows that the multilayer neural network works better than linear regression and persistence (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 it deals with each time series separately without considering spatial dependency among them.

Some hybrid models are proposed to handle linear and non-linear patterns in air quality time series data. [7] presents a hybrid model combining ARIMA and ANN to improve forecast accuracy for an area with limited air quality and meteorological data. The idea is utilizing an ARIMA model to forecast daily maximum PM_{10} moving average, and then an ANN model is used to describe the residuals from the ARIMA model. The result reports the hybrid model outperforms the individual ARIMA model and ANN model. Similarly, in [3], it employs the combination of ARIMA and a non-linear model and observes an improvement on the hybrid model than separate ones with a reduction of 26.31% and 21.05%. However, those hybrid approaches remain the spatial dependency problem in the air quality data. In [31], the authors proposed a hybrid model to forecast the air quality over next 48 hours (i.e., real-valued AQIs for next 6 hours and a max-min range of AQI for next 7-12, 12-24 and 24-48 hours) for each monitoring station. The hybrid model can handle temporal and spatial dependency in separate models (i.e., temporal predictor and spatial predictor) and aggregate two predictors with a Regression Tree. The result shows the hybrid model outperforms individual models. Some fixed-size buffers are set to select the neighborhood stations around the target station. However, geographic distances cannot reflect a proper similarity between two sensors when they are far apart from each other. Moreover, separate models might lose information from each other. A better idea is to describe the spatial dependency among sensor networks more intuitively and model spatiotemporal dependency simultaneously.

Deep learning approaches deliver new promise for 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, e.g., speech recognition [26], traffic forecasting [28]. To jointly model the spatial dependency and the temporal dependency among time series, Li et al. [17] proposes the Diffusion Convolutional Recurrent Neural Network (DCRNN) which combines diffusion convolution with RNN.

In this paper our approach uses DCRNN model to capture both the spatial and temporal dependencies in one model instead of just handling temporal dependency or dealing with them in separate models. The model is able to generate the following 24-hour air quality forecasting results with previous-hour air quality and

meteorological data. To manipulate the spatial dependency, we automatically select crucial geographic features in neighbors that have a great impact on $PM_{2.5}$ concentrations. We construct the graph by computing the similarity of those important features between sensors instead of using geographic distance, which is reasonable to tell the spatial correlation in air quality time series data.

6 CONCLUSIONS AND FUTURE WORK

This paper presented a data driven approach to forecast $PM_{2.5}$ concentrations with previous-hour air quality data and meteorological data. The advantages of our approach include 1) our model could handle both spatial and temporal dependency in the time series data simultaneously and achieved a better performance than other traditional methods; 2) we represent the spatial correlation in a graph with automatically selected important geographic feature types that largely effect $PM_{2.5}$ concentrations and use those important geographic feature types to compute the adjacency graph for 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 air quality-related features for handling other temporal dynamics, e.g., workday/weekend and seasonal effects.

ACKNOWLEDGMENTS

This work is supported in part by the NIH grant 1U24EB021996-01 and by NVIDIA Corporation.

REFERENCES

[1] Martín Abadi et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).
[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, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv: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] KI Hoi, KV Yuen, and KM 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 VK 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, Nur Arina Bazilah Kamisan, et al. 2012. Seasonal ARIMA for forecasting air pollution index: A case study. *American Journal of Applied Sciences* 9, 4 (2012), 570–578.
[17] 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 (ICLR)*.
[18] 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*. ACM, 25.
[19] 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.
[20] World Health Organization, UNAIDS, et al. 2006. *Air quality guidelines: global update 2005*. World Health Organization.
[21] 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.
[22] Patricio Perez and Giovanni Salini. 2008. PM2.5 forecasting in a large city: comparison of three methods. *Atmospheric Environment* 42, 35 (2008), 8219–8224.
[23] 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.
[24] 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.
[25] David YH 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.
[26] 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.
[27] 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.
[28] 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 (SDM)*.
[29] 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.
[30] 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.
[31] 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*. ACM, 2267–2276.