M.S. THESIS

# SPATIOTEMPORAL DEEP LEARNING MODEL FOR CITYWIDE AIR POLLUTION INTERPOLATION AND PREDICTION

SPATIOTEMPORAL DEEP LEARNING 모델을
기반으로 한 도시 전역의 대기 오염 보간과 예측

BY

LE VAN DUC

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

# SPATIOTEMPORAL DEEP LEARNING MODEL FOR CITYWIDE AIR POLLUTION INTERPOLATION AND PREDICTION

SPATIOTEMPORAL DEEP LEARNING 모델을 기반으로 한 도시 전역의 대기 오염 보간과 예측

BY

LE VAN DUC

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

# SPATIOTEMPORAL DEEP LEARNING MODEL FOR CITYWIDE AIR POLLUTION INTERPOLATION AND PREDICTION

## SPATIOTEMPORAL DEEP LEARNING 모델을 기반으로 한 도시 전역의 대기 오염 보간과 예측

지도교수 차 상 균

이 논문을 공학석사 학위논문으로 제출함

2019년 2월

서울대학교 대학원

전기 · 정보공 학부

르 반 득

르반득의 공학석사 학위 논문을 인준함

2019년 2월

위 원 장: 정교민

부위원장: 차상균

위 원: 양인순

# Abstract

Air pollution is one of the most concerns of big cities. Many countries in the world have constructed air quality monitoring stations around major cities to collect air pollutants and make the warning to urban citizens about the air pollution around them. However, air pollution is not uniform in the city, but it is a spatiotemporal problem. It changes by locations (spatial feature) and by time (temporal feature). Consequently, citywide air pollution interpolation and prediction is a requirement of urban people to know the air quality through time and spaces to eliminate the health risks. Moreover, air pollution is affected by many spatiotemporal factors throughout the whole city. Among them, meteorology is recognized to be one the most significant effects to air pollution. Besides that, traffic volume reflects the density of vehicles on roads which is the primary cause of air pollution. Average driving speed indicates the traffic congestion which also reasonably influences air pollution over the city. Finally, external air pollution sources from outside areas are claimed to be the reason contributing to a city's air pollution problem. In this thesis, we present many spatiotemporal datasets collected over Seoul city, Korea such as air pollution data, meteorological data, traffic volume, average driving speed, and air pollution of 3 China areas like Beijing, Shanghai, Shandong, which are known to have the effect to Seoul's air pollution.

Recent research in air pollution has tried to build models to predict air pollution by locations and in the future time. Nonetheless, they mostly focused on predicting air pollution in discrete locations or used hand-crafted spatial and temporal features. Recently, Deep learning models such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long-Short Term Memory (LSTM) are known to be superior in spatial and temporal relating problems. In this thesis, we propose the usage of Convolutional Long-Short Term Memory (ConvLSTM) model, a combination of CNN and LSTM, which efficiently manipulates the spatial and temporal features of the data and outperforms other recent research.

**keywords**: Air pollution, Interpolation, Prediction, Citywide, Spatiotemporal, Deep Learning

**student number**: 2016-27885

# Contents

# List of Tables

# List of Figures

# Chapter 1

# INTRODUCTION

## 1.1 Air pollution description

Outdoor air pollution is now threatening seriously to the human health and life in big cities, especially to elderly and children [Kampa]. This is not a private problem of one country but a global problem. Therefore, many countries in the world have constructed air pollution monitoring stations around major cities to observe air pollutants such as PM2.5, PM10, CO, NO2, SO2 [Wiki] and alert to their citizens if there is a pollution index which excesses the country-specific quality threshold. This section describes these air pollutants and the primary emission sources. PM2.5 is fine atmospheric particulate matter (PM) that have a diameter of less than 2.5 micrometers. PM10 is coarse particulate that is 10 micrometers or less in diameter. From [EPA], PM2.5 and PM10 are emitted directly from some sources, such as construction sites, unpaved roads, fields, smokestacks or fires. Moreover, particles form in the atmosphere as a result of complex reactions of chemicals such as sulfur dioxide and nitrogen oxides, which are pollutants emitted from power plants, industries, and automobiles. CO refers to Carbon Monoxide which is a product of combustion of fuel such as natural gas, coal or wood. Vehicular exhaust contributes to the majority of carbon monoxide let into our atmosphere. NO2 refers to Nitrogen Oxides, expelled from high-temperature combustion.

NO2 forms from emissions from cars, trucks and buses, power plants, and off-road equipment. SO2 is Sulfur Oxides, produced by volcanoes and in various industrial processes. Coal and petroleum often contain sulfur compounds, and their combustion generates sulfur dioxide [Wiki]. As a result, air pollutants are emitted from many sources but one of the common reasons are transportation.

## 1.2 Citywide Air pollution Interpolation and Prediction

Air pollution prediction has emerged as an active research field recently. Much recent research has pointed out that urban air pollution has both temporal and spatial features as in [Wong], [Li], [Le] and so on. It means that air pollution values do not only change time by time but also differ between different locations in a city. In figure 1, we show the air pollution (PM10) values by the hour in 2 monitoring stations in Seoul in January 2017. Two monitoring stations are far apart in locations, one in the west of Seoul and one in the east. We can see that the air pollution values are changed continually hour by hour with the maximum value can reach to more than 270 $\mu$g/m$^3$ and the minimum value can down to less than 20 $\mu$g/m$^3$. Moreover, although 2 mentioned stations are both located in Seoul city, the air pollution values indicated by them differ a lot in some periods. As in figure 1, in some hours around 100, the air pollution value of one station can be 3 times larger than values of another (150 vs. 50). In the paper [Zheng2], the authors researched the spatiotemporal features of air pollution in Beijing, China and also discovered similar trends. The reason for these observations is air pollution depends on a number of factors both by time and by locations. The first one is meteorological factors, which also change in spatiotemporal form. The temperature, humidity, raining of different locations and the wind speed, wind direction make air pollution change from locations to locations. Another critical reason for air pollution is the traffic volume and traffic congestion. The locations with more traffic volume or frequent traffic jam occurring will have around air pollution may be worse. One indication of the traffic jam is the average driving speed on each road, in which a small average speed means there might be traffic congestion. The monitoring stations could help us to have a measurement of air pollution at and around the located points but not for the whole city. For example, in Seoul, we only have 37 monitoring stations covering the area of 600 km2. Consequently, we need to interpolate the air pollution in areas that do not have observation stations nearby. The more accurate interpolation model we could build, the more chances for urban citizens to manage their urban life better. The ability to interpolate and forecast air pollution for any locations in the city and in some time ahead is the citywide Air pollution Interpolation and Prediction function. This function will be the necessary function of any Air pollution control system for Urban areas.

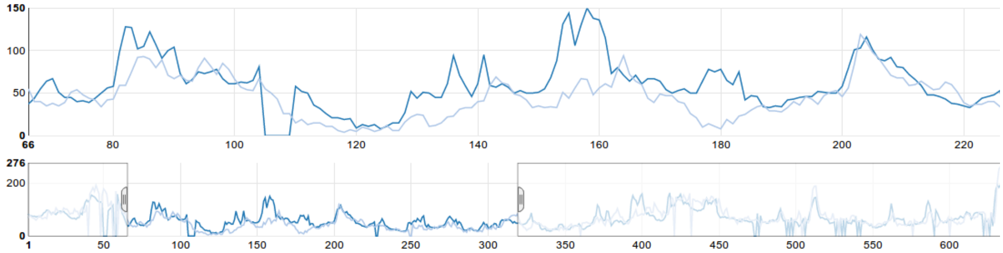


Figure 1: Air pollution (PM10) in 2 locations in Seoul in January 2017. Below is all air pollution and above is a focused, specific time period.

## 1.3 Spatiotemporal datasets introduction

As described earlier, the air pollution changes in spatiotemporal form and we need to interpolate and forecast air pollution in the citywide scale. For this thesis, we have already collected and used many spatiotemporal datasets, specific to Seoul city of Korea. The period of data time is 3 years, from 2015 to 2017. In summary, we already recovered hourly air pollution data of 39 monitoring stations, hourly meteorological data of 28 observation stations, hourly traffic volume data for about 145 main roads in Seoul, and hourly average driving speed in more than 4000 speed-surveying points. Moreover, a recent report from [NIER-NASA] has shown the influence of outside air pollution sources from China to Seoul. To mimic these effects, we have gathered air pollution of 3 areas in China like Beijing, Shanghai, and Shandong from 2015 to 2017. The detail description of each dataset as follows.

The hourly air pollution dataset is quite common in recent research for air pollution prediction problem. Seoul government has constructed 39 air pollution monitoring stations to hourly collect air pollutants such as PM10, PM2.5, Ozone gas (O3), NO2, CO, and SO2. The measurement unit for PM10 and PM2.5 is $\mu$g/m³ and for other pollutants is ppm (parts per million). The stations spread out for all 25 districts in Seoul (see figure 2). Totally we have 24 hours * 3 years * 39 stations (but 2017 only has the data until 09/30) is 937,872 rows. Each row contains the date and time, station address, and the values of 6 air pollutants. Not all stations collect all 6 air pollution components. Instead, the PM2.5 pollution is presented in only 25 stations. In table 1, we present the analysis statistic of the air pollution data.

Table 1: Analysis statistic of Air pollution data

| | SO2 | CO | O3 | NO2 | PM10 | PM2.5 |
|---|---|---|---|---|---|---|
| count | 913,320 | 912,951 | 913,896 | 912,923 | 909,477 | 578,694 |
| mean | 0.00527 | 0.56261 | 0.02150 | 0.03726 | 49.0 | 25.0 |
| min | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 1.0 | 1.0 |
| max | 0.04700 | 4.10000 | 0.17800 | 0.34000 | 1160.0 | 175.0 |

The hourly meteorological data is also popular among air pollution prediction research. In this thesis, we got the meteorological data from the Korea Meteorological Administration agency. Currently, there are total 28 meteorological observation stations in Seoul which observe following information of weather per hour: temperature (Celsius degree), wind direction (degree), wind speed (m/s), precipitation (mm), lowest air pressure (hPa), highest air pressure (hPa), humidity (%). The locations of all stations are illustrated on the map in figure 2. We can see that at least one district have one observation station. The analysis statistic of meteorological data is shown in table 2.

Table 2: Analysis statistic of Meteorological data

| | Temperature | Wind speed | Precipitation | Lowest air pressure | Highest air pressure | Humidity |
|---|---|---|---|---|---|---|
| count | 734,891 | 734,740 | 730,928 | 55,418 | 55,418 | 475,265 |
| mean | 13.5 | 1.64 | 0.1 | 1012 | 1015 | 60 |
| min | -25.2 | 0.00 | 0.0 | 1005 | 1008 | 0 |
| max | 37.6 | 11.20 | 63.0 | 1035 | 1037 | 99.9 |

The traffic volume of main roads is a new dataset within all known air pollution-related research. The Seoul Metropolitan Government has installed the vehicles detector at many survey points and collected the traffic volume every hour. Figure 3 shows the locations of these checkpoints on major roads in Seoul map. There are 4 types of analysis roads in Seoul: the inner roads (도 심): 24, the border roads (시 계): 22 (in 2015), 36 (in 2016 and 2017), the crossroads (간 선): 32 (in 2015), 54 (in 2016 and 2017), the bridge roads (교 량): 12 (in 2015), 22 (in 2016 and 2017), the city highways (도시고속): 0 (in 2015), 9 (in 2016 and 2017). Totally, the number of analysis roads in 2015 is 90 and in 2016 and 2017 is 145 roads. The collected data has a column shows the inflow or outflow direction along the survey road and 24 columns which contain the traffic volume for each hour in a day. There are many statistics from the Seoul Government's annual report, but in this thesis, we only focus on the general statistic of the data as shown in table 3.

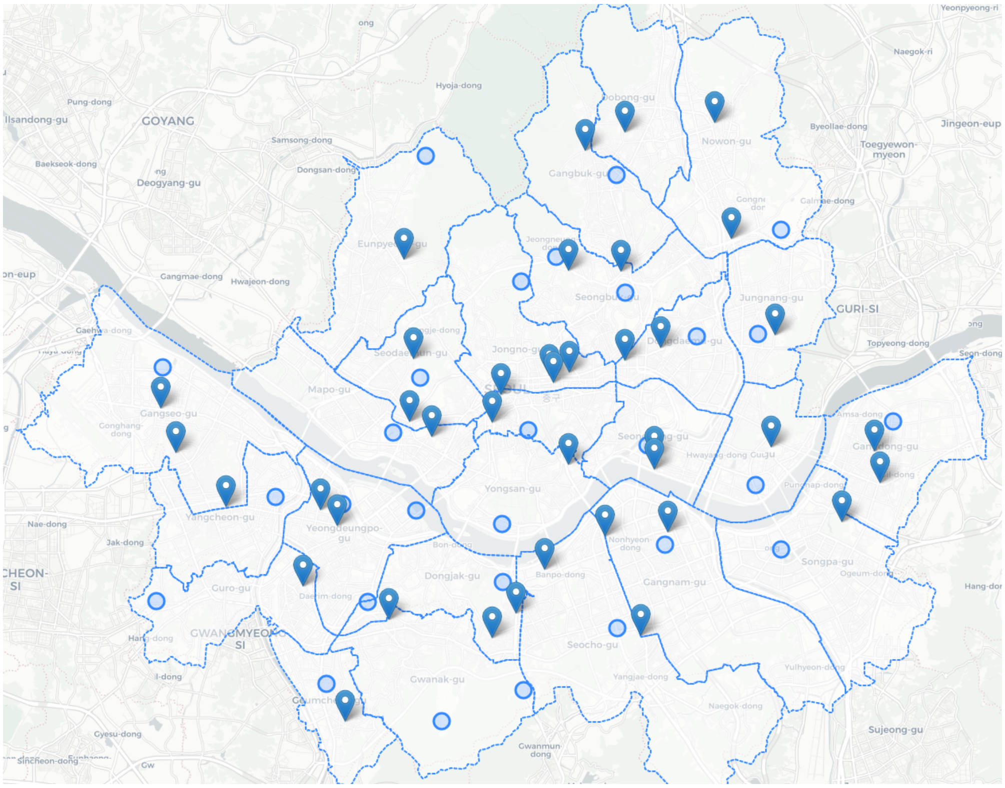


Figure 2: The location of Air pollution monitoring stations (markers) and Meteorological observation stations (circle markers) in Seoul.

Table 3: Analysis statistic of Traffic volume data

| | Traffic volume information |
|---|---|
| count | 4,697,888 rows |
| mean | 1,510 (turns/hr) |
| min | 638 (turns/hr) |
| max | 38,908 (turns/hr) |

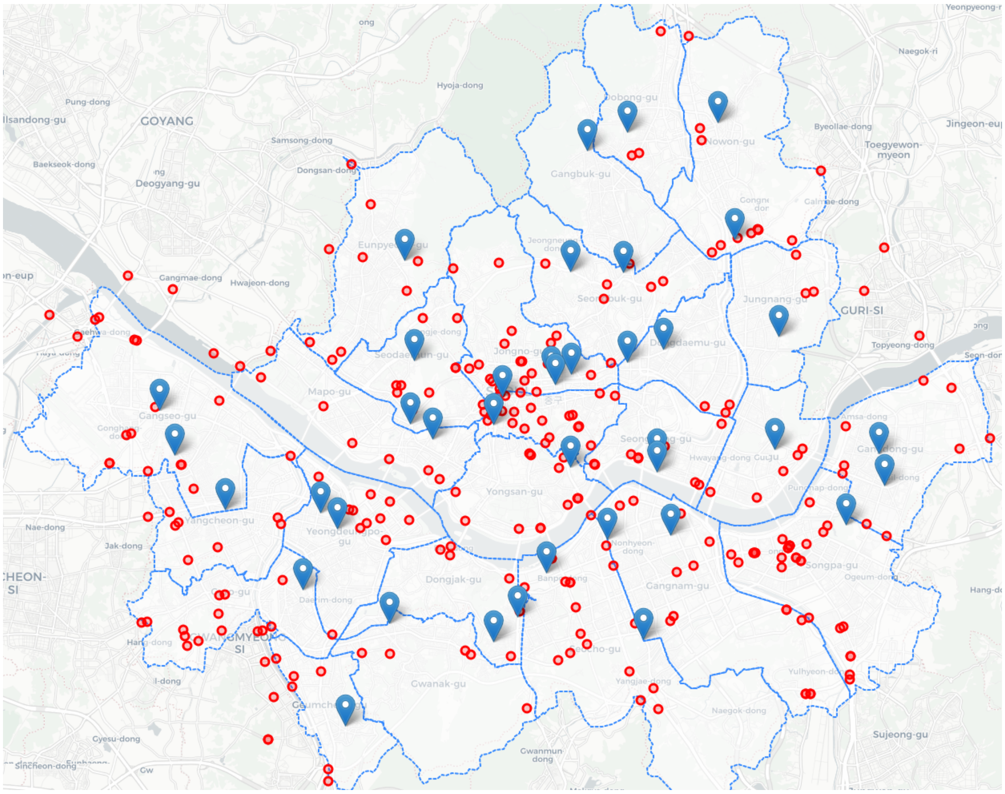


Figure 3: The location of traffic volume survey points (small circles) on major roads in Seoul.

The average driving speed data is also a novel air pollution relating dataset. The Seoul Government has investigated the speed of general vehicles and buses for all primary roads in Seoul. The car roads in the analysis are 487 roads covering the length of 1,434.5 km. The analyzed bus routes are 364. The data is collected in each hour. In figure 4, we show the locations of all survey points of vehicles speed. As we can see, the speed checkpoints are dense and cover quite good the area of Seoul map compared to the air pollution monitoring stations (in markers). The analysis statistic of vehicles speed data is in table 4.

Table 4: Analysis statistic of Average driving speed data

| | Average driving speed information |
|---|---|
| count | 102,453,700 rows |
| mean | 29.6 (km/h) |
| min | 0.6 (km/h) |
| max | 308 (km/h) |

The last collected dataset in this thesis is PM2.5 air pollution from 3 areas in China as Beijing, Shanghai, and Shandong. The data is collected from Berkeley Earth research website (http://berkeleyearth.lbl.gov/air-quality/local/China) and is crossed check with PM2.5 data from the US. Department of State Air Quality Monitoring Program (for Beijing and Shanghai data). The data is also collected hourly and has a total of 78,912 rows.

## 1.4 Thesis contributions

This thesis has three (3) main contributions. Firstly, we claim that citywide Air pollution Interpolation and Prediction is an indispensable function for any Air pollution control system. Secondly, we introduce many spatiotemporal datasets which relate to

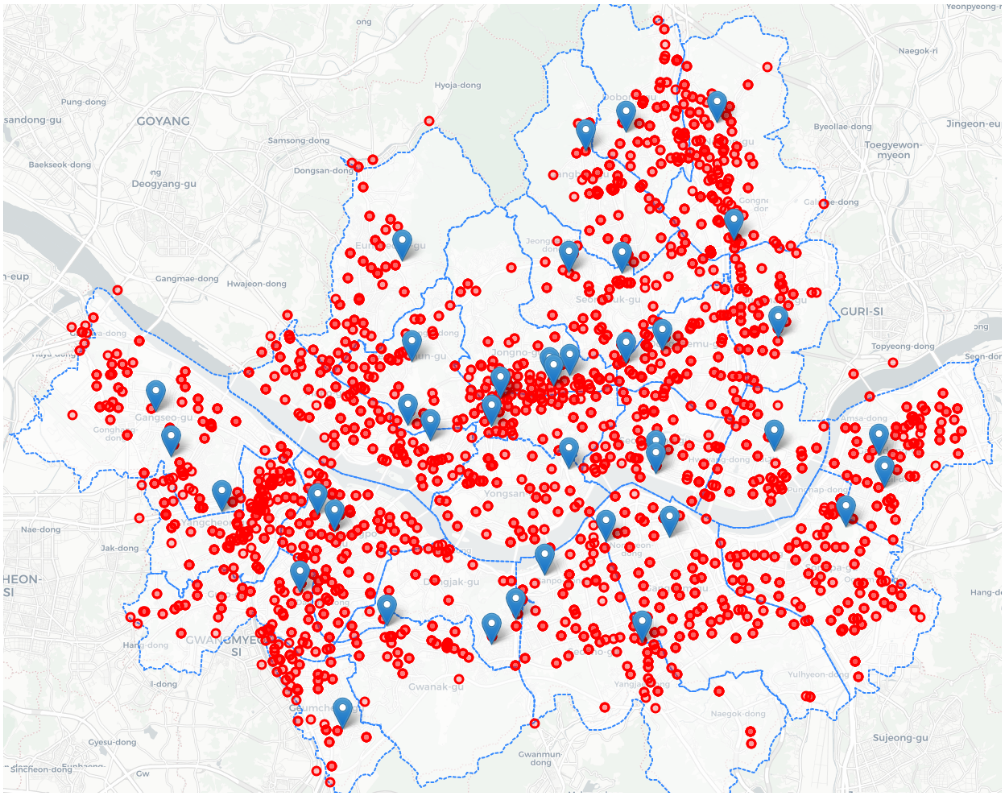


Figure 4: The location of driving speed survey points on all roads in Seoul (small circles).

air pollution. The last contribution is that we present a Deep Learning based Spatiotemporal prediction model for Air pollution. Recently, Deep Learning based algorithms such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) have many successes on spatial and temporal related problems such as image classification, object detection, sequence to sequence prediction and so on. Spatiotemporal air pollution data has both spatial and temporal features, and naturally, CNN and RNN based models are suitable for this problem. For RNN, a more prosperous and more common used variation is Long-Short Term Memory (LSTM) model. In [Shi], the authors proposed a novel combination model of CNN and LSTM called Convolutional LSTM (ConvLSTM) in predicting precipitation satellite images. In this research, we leverage the using of ConvLSTM for Air pollution Interpolation and Forecasting problem with input data of spatiotemporal datasets. ConvLSTM helps us to process the spatial and temporal features of input data at the same time and automatically, surpassing recent research which much relied on hand-crafted spatial and temporal features.

# Chapter 2

# RELATED WORK

Urban Air pollution research has been started for a long time. In paper [Mage], the authors have talked about the history of the urban air pollution problem. In 1972, the UN Conference on the Environment in Stockholm stressed finding solutions for all global environmental pollution problems. In 1974, The United Nations Environment Programme (UNEP) and World Health Organization (WHO) collaborated in the initiation of a Global Environment Monitoring System (GEMS) urban air pollution monitoring network (GEMS/Air). Nowadays, urban air pollution is one of the most worries for any cities in the world, especially for big cities including Seoul, Korea. Urban Air pollution prediction has emerged active research to better control air quality and protect city people's health. In this chapter, we present some of the most related research to the thesis's content.

## 2.1 Spatiotemporal Air pollution interpolation

In this section, we will introduce some Air pollution interpolation research which tried to interpolate Air pollution at locations where are lack of monitoring stations.

In paper [Wong], the authors tried to compare spatial interpolation methods for estimating air quality data. Their used data was the US ambient air pollutants dataset of

6 components like PM10, O3, CO, NO2, SO2, and Lead (Pb). They proposed the usage of 4 basic interpolation models. The first was Spatial averaging which was proposed by Schwartz in 1989. In here, 10 miles was used as the neighborhood distance to average. The second one was the Nearest neighbor method which assigned the air concentration level of the monitoring station nearest to its centroid. The third algorithm was Inverse distance weighting (IDW), in which interpolation weights were computed as a function of the distance between observed sample sites and the site at which the prediction had to be made. And the last method was Kriging, which used the Gaussian process to compute weights, minimizing the variance in the estimated value. Among 4 methods, the authors claimed that Kriging might be more suitable for chosen Air pollution dataset. Regarding our evaluation, these are basic and simple interpolation algorithms which often used as baselines for more advanced/complex methods.

In paper [Li], the authors investigated Spatiotemporal Interpolation methods for Air pollution exposure health problems. The dataset was the daily US PM2.5 air pollution data in 2009. They used Shape Function (SF) based spatiotemporal interpolation method focusing on spatiotemporal interpolation problems in the domain of 2-D space (x, y) and 1-D time (z=t). In the result part, they claimed that SF methods better than IDW and Kriging methods but there were no empirical comparisons supplied. By our evaluation, this is another basic baseline for Spatial air pollution interpolation inspired from Geographic Information System (GIS). Nevertheless, no empirical results in comparison with other methods.

## 2.2 Machine Learning/Neural Networks based Air pollution prediction models

In this section, we mention a number of recent research which used Machine Learning/Neural Networks based models in predicting Air pollution. The common point of this research is both using China air pollution datasets.

In paper [Zheng1], the authors proposed a model called U-Air, in which they tried to infer air pollution values in a grid-based map. Their used datasets are 5 data sources consisting of the Point-Of-Interests (POIs); Road networks like highway and city roads length; Meteorological data; air quality records of Beijing and Shanghai, and the last is the GPS trajectories generated by over 30,000 taxis in Beijing to analysis travel speed, human activities. Their presented model was a co-training-based semi-supervised learning approach, which leverages unlabeled data to improve the inference accuracy. For detail, they built 2 separated classifiers called Spatial Classifier and Temporal Classifier to classify Air pollution value into Air Quality Index (AQI) level like Good, Moderate, Unhealthy, and Hazard.

The newer paper [Zheng2] also emphasized Air pollution Big Data but tried to forecast fine-grained air quality. They used dataset was China air quality dataset of 2,296 stations in 302 cities in China from 8/2012 to 5/2015. The Meteorological data was 3,514 locations, consisting of rain levels, temperature, humidity, wind speed, wind direction. They also used Weather forecasting data with a 3-hour interval of the next 3 days. Their model comprised of four major components: a linear regression-based temporal predictor, a neural network-based spatial predictor to model spatial factors, a dynamic aggregator (Regression Tree) combining the predictions of the spatial and temporal predictors according to meteorological data, and an inflection predictor to capture sudden changes in air quality, such as sudden drop instances from historical data. The model predicted for 1-6 hours (hourly) and min-max values for a 3-time interval: 7-12, 13-24, 25-48 hours ahead. It predicted ΔAQI (not AQI itself).

The paper by [Hsieh] stated a new problem in urban air quality control. Their objective was to suggest locations in a city to build new monitoring stations to get the most efficient performance. The dataset was Air Quality Records of Beijing dataset, PM2.5 + PM10, 22 stations, 8/2012 10/2013. Missing data were treated as unobserved data to infer. Other datasets were the Meteorological Data in Beijing, hourly; POIs data of categories and density, with 12 POI types; and Road networks data. In their model, to infer AQI values, they divided Beijing city into disjointed grids of 1km*1km. They proposed a semi-supervised learning algorithm including 4 stages. Regard to recommend locations for building new monitoring stations, they suggested Greedy-based Entropy Minimization (GEM) algorithm which aims at ranking locations based on their capability to reduce uncertainty.

Another surveyed papers are 2 new papers in 2018. The first one [Qi] is quite similar to our approach in this thesis. In [Qi], the authors proposed a model named Deep Air Learning (DAL) which was used to interpolate, predict and analyze input features for fine-grained air quality. The authors also used data of Air pollution and Meteorological data of Beijing city. They introduced Spatiotemporal Semi-supervised Learning in Neural Network which used both labeled and unlabeled data to interpolate and predict a grid's pollution value. Loss function was a spatial and temporal loss value of neighbor grid-cells. They manually chose training features of the size of spatial and temporal neighbors (e.g., chosen as 2).

The second paper [Cheng] has leveraged the usage of Attention Model in Urban Air Pollution problem by learning the weights of monitoring stations in inferring air pollution from neighbor stations dynamically. About the model, they suggested a generic neural attention model, named ADAIN (Attentional Deep Air quality Inference Network), for spatially fine-grained urban air quality inference. They explored the using of deep neural networks (DNNs) for modeling heterogeneous data in a unified way, and learning complex feature interactions without expensive handcrafted feature engineering.

Regarding our evaluation of above mentioned related work, some research proposed grid-based air pollution interpolation or prediction. Nevertheless, they only focused on discrete locations, not considering the whole city to be an image as in our approach. Furthermore, they used much hand-crafted spatial and temporal features which were difficult to generalize to other similar problems.

## 2.3 Spatiotemporal Deep Learning models

In this section, we survey general Spatiotemporal Deep Learning algorithms. In [Shi], the authors have proposed a Convolutional LSTM (ConvLSTM) model and used for precipitation forecasting. In the paper, the authors have discussed a number of models for forecasting spatiotemporal problems. Fully Connected LSTM (FC-LSTM) is a common LSTM architecture which uses full connections in input-to-state and state-to-state transitions, no spatial information is encoded. Therefore, this model have difficulty in predicting spatiotemporal values. ConvLSTM uses convolution operators in both state-to-state and input-to-state transitions, leverages both spatial and temporal features in the input data. As a result, ConvLSTM is suitable for the spatiotemporal problem. In the paper, the authors also demonstrated that ConvLSTM was better than FC-LSTM in spatiotemporal problems like moving MNIST and weather radar echo images of Hong Kong for precipitation forecasting.

The spatiotemporal problem is also fit for crowd flows prediction problem. In [Zhang2], the authors presented a Deep Neural Network (DNN) Spatiotemporal (DeepST) for predicting Crowd flows in Beijing and New York. They proposed the DeepST model based on Convolutional Neural Network for 3 sequences of data: 1) temporal closeness; 2) period; 3) seasonal trend. Residual Units (as in ResNet) were used to leverage very deep network to capture more citywide dependencies. And the last layers were Fusion layers to combine deep network results with external factors (such as meteorology, holidays).

Zhongjian et al. in a paper from JICAI 2018 has proposed a model named LC-RNN for Traffic Speed Prediction [Zhongjian]. Their model consisted of a Lookup Convolution Layer to extract spatial information of road networks and some LSTM layers to learn temporal information with a fusion layer to combine other traffic speed's periodcity and context extraction information with LC-RNN's output. Similar ConvLSTM, their model also was a combination of CNN and LSTM but in 2 separate steps.

# Chapter 3

# SPATIOTEMPORAL DEEP LEARNING MODEL

In this chapter, we present our proposed model for citywide Air pollution Interpolation and Prediction based on Spatiotemporal Deep Learning. Firstly, we talk about CNN and LSTM models which are proved working efficiently with spatial and temporal problems. Next, we propose the usage of ConvLSTM which is the combination of CNN and LSTM and claim its suitability for spatiotemporal Air pollution problem. Finally, in the last section, we show the complete Spatiotemporal Deep Learning model for our citywide Air Pollution Interpolation and Prediction.

## 3.1 CNN and LSTM models

Convolutional Neural Networks (CNN) is one of the most successful Deep Learning algorithms, especially in image classification, object detection. Some of the most well-known CNN models are AlexNet (2012), ZFNet (2013), GoogleNet/Inception (2014), VGGNet (2014), and ResNet (2015). In figure 5, the architecture of AlexNet show us the fundamental modules of a CNN model. A CNN model typically consists of many Convolution layers to extract features from the input image, many Pooling layers to reduce the output size and make the filter more robust, some dropout layers for regularization and one or some fully connected layers at last to produce the final output.

The input to a CNN is usually an image with 3 dimensions: width, height, and depth (or channel). If the image channel is 3, then we have a Red-Green-Blue (RGB) image. Alternatively, if the channel is 1, then we have a gray-scale image. The most important layer for a CNN model is Convolution layer which helps extract spatial features from image input. Convolution layer uses a convolution operator which keep the spatial relationship between image pixels. In the convolution layer, we have a set of learnable filters. Each filter is small spatially along width and height but extends through the full depth of the input volume. For example, with a typical filter of size 3×3×3 (it means the width and height of the filter is 3, and the depth is 3 because the image input has 3 channels), we will slide (or convolve) each filter across the width and height of the input and compute the dot products between the entries of the filter and the input at any position. The output is then activated by a non-linear activation function such as sigmoid, Rectifier Linear Unit (ReLU) or tanh and make an activation map. We will stack these activation maps along the depth dimension and make the output volume. The described convolution operator allows CNN to identify spatial patterns of the input image such as edges, shading changes, shapes, objects, and so on. In figure 6, also from AlexNet paper ([Kriz]), the authors showed the spatial features learned by Convolution filters. Back to our Air pollution Interpolation problem, we need to predict air pollution for any locations throughout a city. If we divide the city map into a grid and consider each grid-cell a pixel of an image then we will have an image with the channel is 1 which means a gray-scale image. As in figure 7, we have already made this transformation for Seoul city by dividing the rectangle which covers the city map into a 32×32 image; each dimension is divided by 32 equal parts.

Long-Short Term Memory (LSTM) is a special kind of Recurrent Neural Network (RNN), which recently works as a standard Deep Learning algorithm for sequence predicting problems like speech recognition, language translation, and so on. The ar-

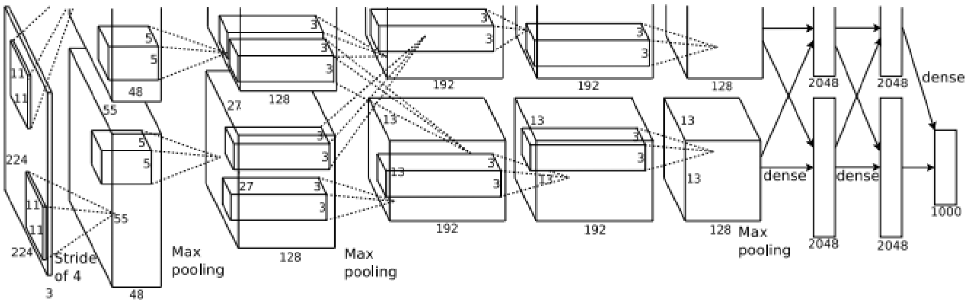


Figure 5: AlexNet Convolutional Neural Network architecture.

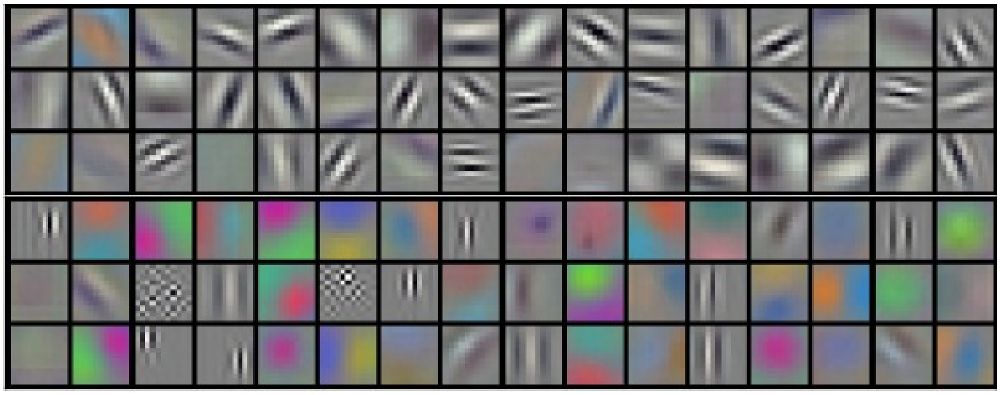


Figure 6: Spatial features from input images learned by CNN filters.

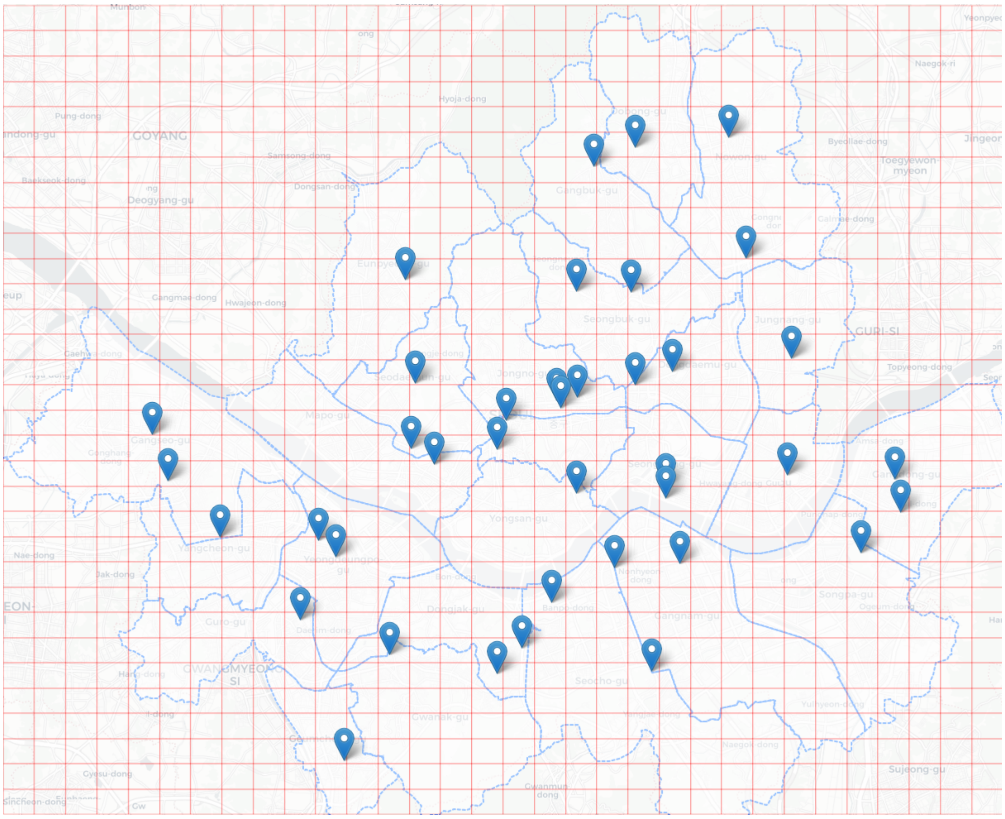


Figure 7: The grid-map of Seoul city (32×32).

chitecture of an LSTM layer is as follows [Colah] and is illustrated in figure 8. At any time t, the input to an LSTM cell is actual data input $x_t$ and the hidden state from previous cell $h_{t-1}$. The first step in an LSTM is to decide which information we are going to output from the cell state. This decision is made by a sigmoid layer called the “forget gate layer”. It looks at $h_{t-1}$ and $x_t$, and outputs a number between 0 and 1 for each number in the cell state $C_{t-1}$. A 1 represents “completely keep this” while a 0 represents “completely get rid of this.”. The equation for the above statement is in equation (1).

$$f_t = (W_f * [h_{t-1}, x_t] + b_f) \tag{1}$$

$f_t$ is the output of the forget gate, $W_f$ and $b_f$ are corresponding weights and biases. * is the matrix-vector multiplication.

The next step is to decide what new information we are going to store in the cell state. This step has two parts. First, a sigmoid layer called the “input gate layer” decides which values we will update. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state. In the next step, we combine these two to create an update to the state. The equations are in equation (2) and (3).

$$i_t = (W_i * [h_{t-1}, x_t] + b_i) \tag{2}$$

$$C_t = tanh(W_C * [h_{t-1}, x_t] + b_C) \tag{3}$$

We multiply the old state by $f_t$, forgetting the things we decided to forget earlier. Then we add $i_t \odot \tilde{C}_t$ with $\odot$ is the Hadamard product or element-wise matrix-matrix multiplication. This is the new candidate values, scaled by how much we decided to update each state value, as shown in equation (4).

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4}$$

Finally, we need to decide what we are going to output. This output will be based on our cell state but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we are going to bring out. Then, we put the cell state through tanh (to push the values to be between 1 and 1) and multiply it by the output of the sigmoid gate, so that we only produce the parts we decided to. The equations in (5) and (6) show these transformations.

$$o_t = (W_o * [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \odot tanh(C_t) \tag{6}$$

The output $o_t$ in (5) and the hidden state $h_t$ in (6) is the output of the current cell, and they will be the inputs of the next cell in the LSTM loop.

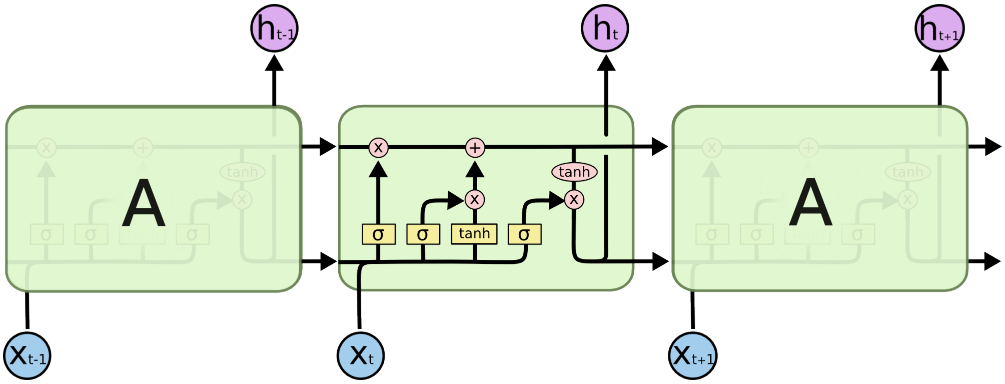


Figure 8: The architecture of a common LSTM layer.

## 3.2 ConvLSTM model

In this section, we introduce our proposal for citywide Air pollution Interpolation and Prediction. As presented in sections above, Urban Air pollution has both spatial and temporal characteristics. Therefore, to efficiently predict air pollution anywhere (interpolation) and at any time (prediction), we need a model which leverages both spatial and temporal features. Moreover, we also stated that CNN and LSTM are 2 Deep Learning models which give high performance on spatial and temporal problems. A combination of both CNN and LSTM model will capture better spatiotemporal features and therefore is suitable for our addressing Air pollution problem. In 2015, X. Shi et al. from Hong Kong University of Science and Technology had proposed a model for precipitation forecasting named Convolutional LSTM Network which was an extension of LSTM model but tried to catch spatial features to have a better prediction on a spatiotemporal problem like precipitation [Shi]. As our Air pollution problem is also spatiotemporally based, we propose to use ConvLSTM for our Air pollution research and claim that this model gives superior performance compared to other solutions. For this section, we describe in detail the ConvLSTM model, and then in the next section, we show how to apply it to Air pollution Interpolation and Prediction.

In [Shi], the input is a spatial region represented by an M x N grid which consists of M rows and N columns. Inside each grid-cell, there are P measurements which change by time. Therefore, the observation at any time can be represented by a tensor X $\in$ $\mathrm{R}^{\mathrm{P \times M \times N}}$, where R denotes the domain of the observed features. If we record the observations periodically, we will get a sequence of tensors $\hat{X}_1$, $\hat{X}_2$,..., $\hat{X}_t$. The spatiotemporal sequence forecasting problem is to predict the most likely length-K sequence in the future given the previous J observations which include the current one, as in equation (7).

$$\tilde{X}_{t+1}, ..., \tilde{X}_{t+K} = \underset{X_{t+1},...,X_{t+K}}{argmax} \; p(X_{t+1}, ..., X_{t+K} | \hat{X}_{t-J+1}, ..., \hat{X}_t) \tag{7}$$

Commonly, LSTM networks are used for a uni-variate variable where we have a single variable input. In this case, we use 6 equations from (1) to (6) introduced in the above section. In the spatiotemporal sequence forecasting, we can see it as a multivariate version of LSTM where the input, cell output, and states are all 1D vectors. In [Shi], the authors called this FC-LSTM (Fully Connected LSTM) with the following equations.

$$f_t = (W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}\odot c_{t-1} + b_f) \tag{8}$$

$$i_t = (W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}\odot c_{t-1} + b_i) \tag{9}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{10}$$

$$o_t = (W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \odot c_t + b_o) \tag{11}$$

$$h_t = o_t \odot tanh(C_t) \tag{12}$$

Although the FC-LSTM layer has proven powerful for handling temporal correlation, it lacks the support for spatial features. To address this problem, [Shi] proposed an extension of FC-LSTM which has convolutional structures in both the input-to-state and state-to-state transitions. All the inputs $\mathrm{X}_1$,..., $\mathrm{X}_\mathrm{t}$, cell outputs $\mathrm{C}_1$,..., $\mathrm{C}_\mathrm{t}$, hidden states $\mathrm{H}_1$,..., $\mathrm{H}_\mathrm{t}$, and gates $\mathrm{i}_\mathrm{t}$, $\mathrm{f}_\mathrm{t}$, $\mathrm{o}_\mathrm{t}$ of the ConvLSTM are 3D tensors whose last two dimensions are spatial dimensions (rows and columns). The ConvLSTM determines the future state of a particular cell in the grid by the inputs and past states of its local neighbors. This can be achieved by using a convolution operator in the state-to-state and input-to-state transitions as shown in figure 9. The equations for ConvLSTM are shown from (13) to (17) with * is now the convolution operator and $\odot$ is still element-wise matrix-matrix multiplication.

$$f_t = (W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f) \tag{13}$$

$$i_t = (W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i) \tag{14}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \tag{15}$$

$$o_t = (W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_t + b_o) \tag{16}$$

$$H_t = o_t \odot tanh(C_t) \tag{17}$$

For the spatiotemporal sequence forecasting problem, [Shi] suggested using the structure shown in figure 10 which consists of two networks, an encoding, and a forecasting network. The initial states and cell outputs of the forecasting network are replicated from the last state of the encoding network. Both networks are formed by stacking several ConvLSTM layers. As the prediction target has the same dimension as the input, to generate the final prediction, all the states in the forecasting network are concatenated and feed them into a 1 × 1 convolution layer. In the next section, we present how we use ConvLSTM for our citywide Air pollution Interpolation and Prediction problem.

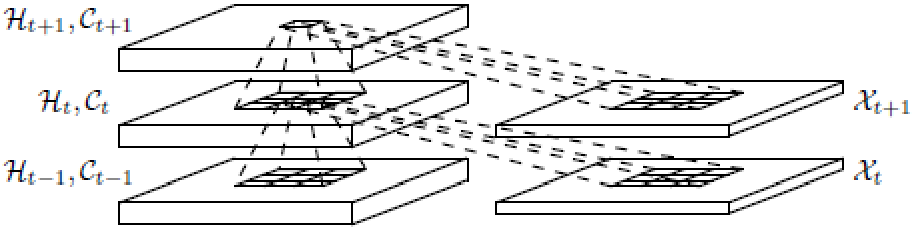

$\mathcal{H}_{t+1}, \mathcal{C}_{t+1}$
$\mathcal{H}_t, \mathcal{C}_t$
$\mathcal{H}_{t-1}, \mathcal{C}_{t-1}$
$\mathcal{X}_{t+1}$
$\mathcal{X}_t$

Figure 9: Inner structure of a ConvLSTM network, taken from [Shi].

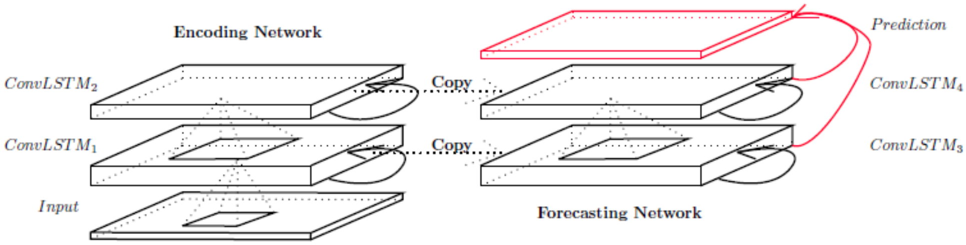


Figure 10: Encoding-Forecasting ConvLSTM structure for spatiotemporal sequence predicting, taken from [Shi].

## 3.3 Air Pollution Interpolation and Prediction

We need to interpolate Air pollution for Everywhere in a city based on the existed monitoring stations. We divide the city' covering rectangle into a grid of width x height cells and assign monitoring stations into grid-cells. The air pollution value in a grid-cell is the aggregated value of all assigned stations' values at a time stamp t. The grid-cell which has no assigned monitoring stations stores value 0. That means we do not have information of air pollution at that point and it will not participate in the training process later. Thus, at any time t, we have a gray-scale image of dimension width x height representing for the city and the pixel values are the aggregated air pollution values at that time. Figure 11 shows these gray-scale images of PM2.5 air pollution for the Seoul city in 3 consecutive hours h = 0, 1, 2. We can see the number of pixels which have the value greater than 0 is quite small compared to zero-value pixels. We need to predict the missing values as good as possible via interpolating. As discussed, the air pollution in a city depends on many factors like meteorology, traffic volume, average driving speed or external air pollution sources. These factors are also represented by the grid map as air pollution. For meteorological data, we assigned the weather observation stations into the corresponding grid-cells, and average values like in air pollution case. For traffic volume and driving speed, the survey point's locations were used to assign them to the grid-cell, and the traffic's volume and speed are also aggregated. With external air pollution sources, because they are cannot be assigned directly to the grid-cell, we embed them into grid-map via pre-training mechanism. Consequently, we have many sequences of "images" which we can apply spatiotemporal Deep Learning model to them. In figure 12, we present the general architecture for our proposal prediction model. With many spatiotemporal input datasets, using our prediction model plus some forecasting datasets such as meteorology or traffic, we can predict air pollution everywhere (citywide scale) and at any time (forecasting).

The spatiotemporal Deep Learning prediction model is a ConvLSTM model as in [Shi] which was described in the previous section. In our case, we do not use patch size to make 3D input images but using gray-scale images as 2D input tensors with MxN dimension. The input tensors are not only air pollution values but are the combination of air pollution values and other influential factors' values at corresponding cells. Denotes $X_a \in {R_a}^{P_a \times M \times N}$ is the air pollution input tensor, where Ra is the air pollution domain, Pa is the range of air pollution values. Similarly, $X_m \in {R_m}^{P_m \times M \times N}$ is the me-

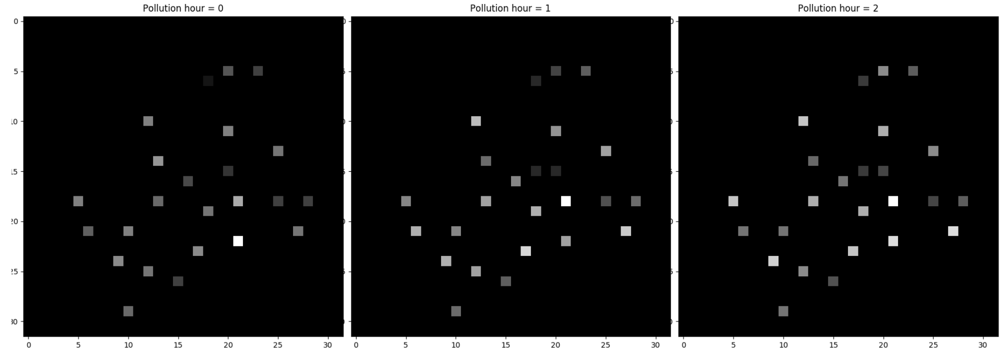


Figure 11: Gray-scale images of PM2.5 air pollution for the Seoul city in 3 consecutive hours h = 0, 1, 2.

teorological input tensor, $X_t \in {R_t}^{P_t \times M \times N}$ is the transportation traffic input tensor, $X_s \in {R_s}^{P_s \times M \times N}$ is the vehicles average speed input tensor, and $X_o \in {R_o}^{P_o \times M \times N}$ is the outside air pollution input tensor. In which $R_m$, $R_t$, $R_s$, and $R_o$ are the meteorological, traffic, speed and outside air pollution domain, respectively, and $P_a$, $P_t$, $P_m$, and $P_o$ are the meteorological, traffic, speed and outside air pollution range of values, respectively. Then the input tensor X of the model is a concatenation of all described input tensors: $X = X_a + X_m + X_t + X_s + X_o$, in which + is a vector concatenation operator. Therefore, for our interpolation and prediction problem, if we want to forecast for K hours, the equation will similar in equation (7).

$$\tilde{X}_{t+1}, ..., \tilde{X}_{t+K} = \underset{X_{t+1},...,X_{t+K}}{argmax}\ p(X_{t+1}, ..., X_{t+K}|\hat{X}_{t-J+1}, ..., \hat{X}_t) \qquad (18)$$

In equation (18), K = 1 is our interpolation and K $>$ 1 is the prediction problem. The complete model is shown in figure 13 where we show how we embed the Outside air pollution sources to be the spatiotemporal input of the model. The output of the

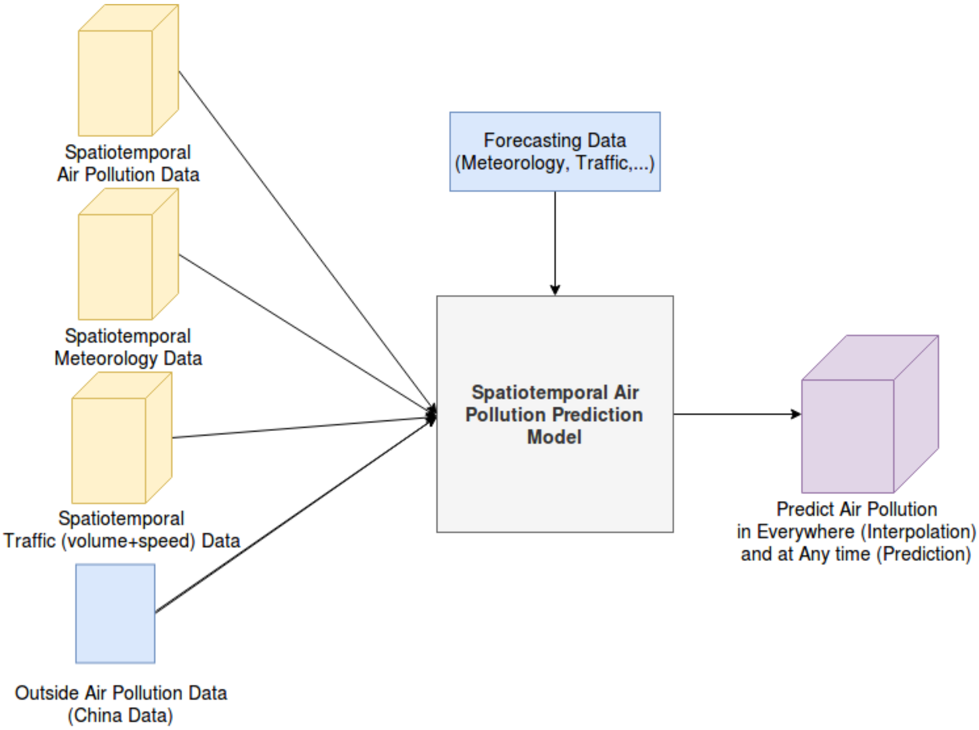


Figure 12: General spatiotemporal prediction model for Air pollution Interpolation and Prediction.

ConvLSTM is then fed into a 1×1 convolution layer to produce the final output. 1×1 convolution is called a “feature pooling” technique where it allows us to sum pooling the features across the depth channel while still keep the spatial characteristic of the feature map. Using 1×1 convolution at the last layer before the output layer, we can transform the ConvLSTM network’s output volume into the final output with the same 2D dimension, but the channel is the out channel of the prediction image. The output also has the grid-based form like the input, and we can use it to determine air pollution everywhere in the city.

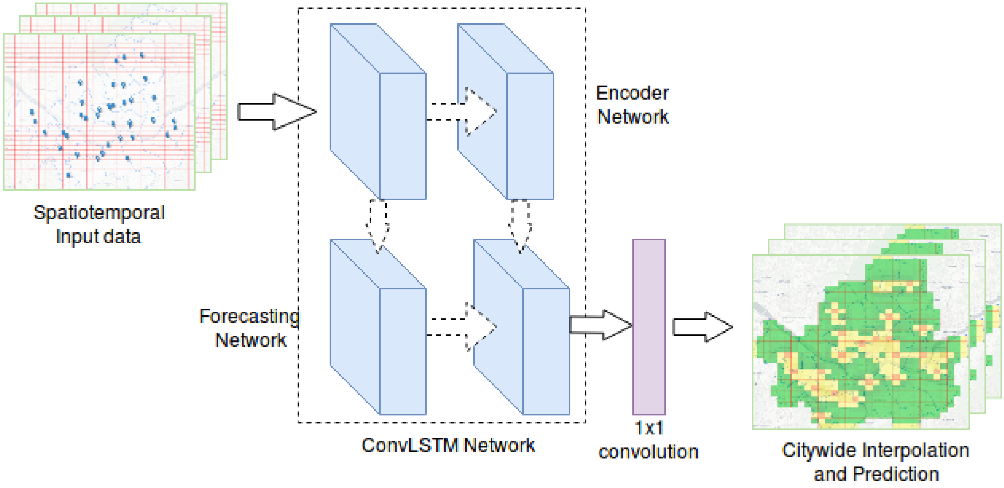


Figure 13: The complete spatiotemporal Deep Learning model for interpolating and predicting Air pollution in a citywide scale.

# Chapter 4

# EXPERIMENTS AND EVALUATIONS

In this chapter, we present our experiments with proposal dataset and model. We also evaluate our model with related baselines to show our superior results. The first section is the baseline description.

## 4.1 Baselines description

Among the recent research which was shown in the related works chapter, Deep Air Learning (DAL) model in [Qi] is the most relevant model to our approach. The authors also divided the studying city (in their case was Beijing) into the grid and tried to interpolate the air pollution in grid-cells which have no monitoring stations information. Those authors also claimed that their model was able to predict air pollution in some time ahead. The most relevant part of their research to ours is that they leveraged the using of spatial and temporal features of the input data. Nevertheless, they still used hand-crafted spatial and temporal features for their model. On the other hand, we use ConvLSTM network, which automatically finds the relationship of the spatial and temporal features while training with the spatiotemporal input data.

In the DAL model, we only interest the spatiotemporal semi-supervised neural network as shown in figure 14. The authors stated that the information contained in unlabeled examples could be utilized to better exploit the geometric structure of the data, especially for the spatiotemporal data. They said that an essential statistical characteristic of spatiotemporal data is that nearby (in space and time) observations tend to be more alike than those far apart. Based on this characteristic, they proposed a novel method which embeds spatiotemporal semi-supervised learning in the output layer of the neural network by minimizing the following loss function between the nearby observations over the labeled and unlabeled training set. The nearby features were chosen manually as 2 for both spatial and temporal neighbors.

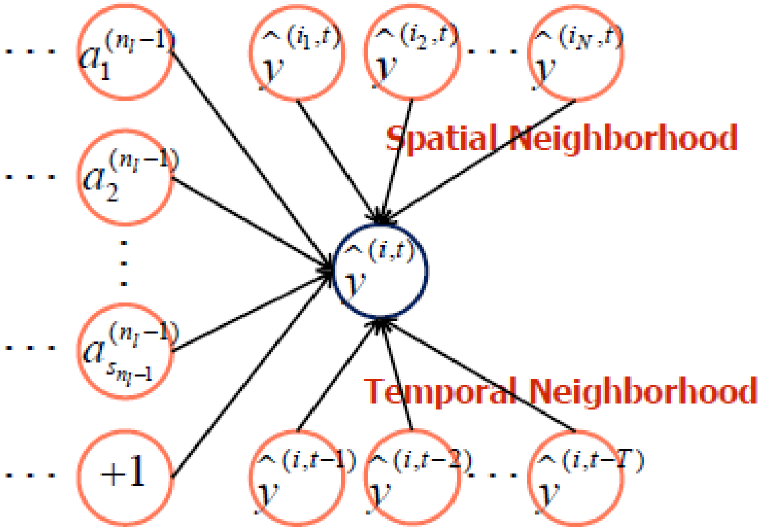
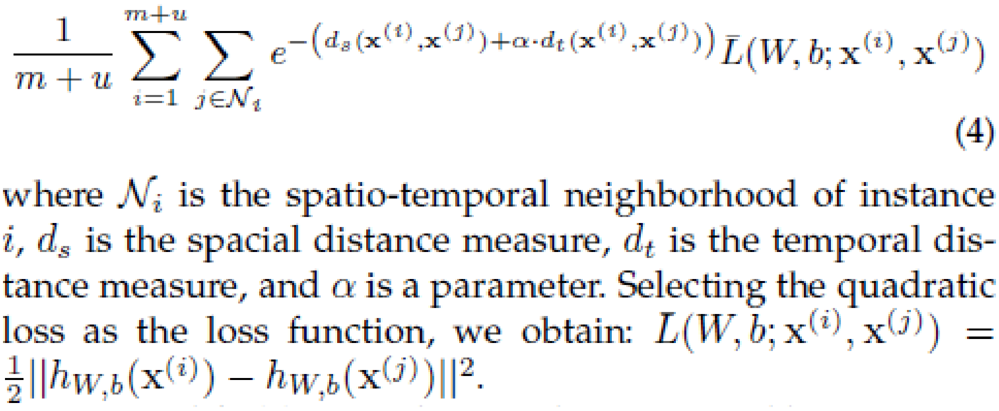


Figure 14: The graph of the spatiotemporal semi-supervised neural network of DAL model from [Qi].

To make this model for our baseline comparison, we re-implemented it for our datasets of Seoul city. In their paper, they used a pre-trained auto-encoder for input data and then tuned with their proposed spatiotemporal loss. We also trained an auto-encoder with 4 layers and used the pre-trained model for next phase training. We implemented DAL for both interpolation and prediction tasks.



$$\frac{1}{m+u}\sum_{i=1}^{m+u}\sum_{j\in\mathcal{N}_i} e^{-\left(d_s(\mathbf{x}^{(i)},\mathbf{x}^{(j)})+\alpha\cdot d_t(\mathbf{x}^{(i)},\mathbf{x}^{(j)})\right)} L(W,b;\mathbf{x}^{(i)},\mathbf{x}^{(j)}) \tag{4}$$

where $\mathcal{N}_i$ is the spatio-temporal neighborhood of instance $i$, $d_s$ is the spacial distance measure, $d_t$ is the temporal distance measure, and $\alpha$ is a parameter. Selecting the quadratic loss as the loss function, we obtain: $L(W,b;\mathbf{x}^{(i)},\mathbf{x}^{(j)}) = \frac{1}{2}||h_{W,b}(\mathbf{x}^{(i)}) - h_{W,b}(\mathbf{x}^{(j)})||^2$.

Figure 15: The loss function and its description of DAL model, taken from [Qi].

In addition to the DAL baseline, we also make other 2 baselines of Deep Learning based models to show how ConvLSTM is better in both spatial and temporal features exploring. The first model in a CNN Encoder-Decoder model which focuses on spatial features learning and the second one is a Stacked FC-LSTM model which is good for temporal features recovering.

## 4.2 Experiments and Evaluations

First of all, we describe how we pre-processed the collected datasets for our experiments. To make these datasets to be the input of our model, we need to translate them by the grid map of Seoul. The Seoul city is covered by the rectangle which has the latitude and longitude coordinates are 37.701 for the maximal latitude (north), 37.435 for the minimal latitude (south), 126.767 for the minimal longitude (west) and 127.812 for the maximal longitude (east). We divide the Seoul city map into 32 cells each direction that means we have a grid map of 32×32 or 1024 cells. As a result, the cell area is approximate 1 square km in the real scale. In following, we illustrate how to fit these datasets into this 32×32 grid map.

The air pollution data has 6 air pollutants for each row, which are SO2, CO, O3, NO2, PM10, and PM25. Each row is represented for an hour and belongs to a monitoring station. We assigned each monitoring station to the corresponding grid-cell by its latitude and longitude coordinates. Some grid-cells are having more than 1 assigned stations and some have no station. The grid-cell with more than 1 stations belonged to have value is the average values of all stations and the grid-cell with no stations keeps storing value 0. Thus, we received a gray-scale image of dimension 32×32 for one hour in 3 years. Because each type of air pollutants has a different distribution, we save 6 datasets of air pollution input and train different models for each dataset inspire of using the same model architecture. In figure 11 we already showed the gray-scale image of PM25 pollution for 3 hours. For all experiments in this thesis, we use only PM2.5 pollution datasets to demonstrate for our proposed model and its results. The grid-based air pollution data is then normalized to the range [0-1] by using Min-Max normalization.

The making of the meteorological grid-based image is similar. We also put weather observation stations into grid-cells based on their latitude and longitude values. With meteorological data, for each row, we have 7 values like temperature, wind speed, wind direction, rainfall, lowest air pressure, highest air pressure, and humidity. We only can aggregate the value of 5 numeric feature like temperature, wind speed, air pressure, and humidity. Wind direction is a categorical feature such as North, South, West-North, and so on. With wind direction, we did not average but chose one of the values if there are many stations put into a grid-cell. Moreover, in contrast with the air pollution data pre-processing mechanism, we do not store value 0 to grid-cell which do not have the station information. Instead, we tried to fill the missing grid-cell by a spatial interpolating method. We chose the nearest neighbor method by interpolating a missing cell by its nearest neighbor which was previously assigned a weather observation station. This method can apply to both numeric and categorical features. The chosen interpolation method was acceptable because following [Beek], the meteorological conditions do not change much in a range of 50 km. The resulting data is then normalized to the range [0-1] by using Min-Max normalization except for wind direction field which we use One-Hot Encoding to encode. The finally processed meteorological data has 21 columns (wind direction field is encoded to a 16-column one-hot encoding vector).

The grid-based transformation for traffic volume and average driving speed is similar to air pollution. The geometric coordinates of each survey point for traffic volume and speed are used to determine its cell in the grid-map. The value is averaged if there is a cell having more than 1 point and if there is a cell having no points then we still keep its value to 0 because we do not have a solid theory as in the meteorological data. Furthermore, it makes sense that for a location which we do not have its data then it will not contribute to the prediction of air pollution at that point. The data is also normalized to the range [0-1] with Min-Max normalization.

The outside air pollution of 3 areas in China is kept untouched because we use an additional model to pre-train their spatiotemporal affection to Seoul air pollution as mentioned in chapter 3.

For experiments and evaluations, we split the datasets into the training set and test set. The training set is 2 years, 2015 and 2016 and the test set is the year 2017. With this splitting mechanism, the training set is quite larger than test set (2 times larger) which helps us to get enough data for training. More important, choosing the training set is 2 years, and test set is 1 remaining year ensure the training and test set have the same distribution and still make our model to have a good generalization. We also split training set into dev set and validation set. The validation set was chosen as 3 last months of 2016 which means 92*24 = 2208 rows. The dev set length is 15,336 rows and the test set length is 6504 rows.

Regarding the forecasting task, we chose to predict for 12 hours. That means we can predict from 1 to 12 hours in the future.

We used Tensorflow Deep Learning framework ([Abadi]) from Google to build the baselines and our model. Tensorflow has supported for Neural Network, CNN, RNN, LSTM, and ConvLSTM network. If not explicitly stated, all experiments in this thesis used the learning rate is 0.001, batch size is 128, training steps are 200, L2 regularization with beta value is 0.01 and the dropout ratio is 0.5. We used Adam optimizer which adapts the learning rate for each parameter by performing smaller updates for frequent parameters and more substantial updates for infrequent parameters for all of our training. The metric for the test set's result is the root mean squared error (RMSE) between the actual air pollution values and the prediction/interpolation values. This is a metric which is commonly used in the regression problem like our Air pollution Interpolation and Prediction. RMSE is only calculated for the pixels which have monitoring stations assigned. If RMSE is smaller then the model's performance is better.

We trained all baselines and our model on a DGX station server with 4 Nvidia Tesla V100 GPU of 16 GB memory each. Using GPU helps us to decrease our training time to less than 5 minutes compare to some hours when using CPU.

### 4.2.1 Air pollution Interpolation: experiments and evaluations

#### DAL interpolation

The DAL interpolation implementation has the input time step is 1 current hour and the output time lag is also 1 hour ahead. The number of Auto-Encoder weights for each layer is 2000. In the paper [Qi], the authors use 2 hyper-parameters called alpha and beta to control the effect of spatial and temporal loss respectively. In their paper, alpha and beta are chosen as 10 and 15 but in our own implementation, we found that the original values did not give very good results so we tested around and determined the values of alpha and beta are 2 and 3 respectively. After training Auto-Encoder model and save to a checkpoint, we restore the pre-trained checkpoint for Spatiotemporal Semi-supervised regression model training. For the spatial and temporal loss, we did not compute the loss separately for each pair of actual and prediction values but we tried to make 2 large tensors by concatenating all actual and all prediction values into each tensor. Then we only need 1 computation to compute the loss for spatial or temporal neighbors. The final loss is the combination of labeled loss, weighted spatial loss and temporal loss of all labeled and unlabeled data. The RMSE result on the test set is shown in table 5.

### ConvLSTM interpolation

The implementation of ConvLSTM interpolation is similar with 1-hour time step as input and 1 hour ahead as interpolation. The number of layers for ConvLSTM network is 1 encoder layer and 1 forecasting layer with the output channels are 64. The kernel size for each encoder and forecasting layer is 3×3. The output size is the grid map size which means 1024. In contrast to DAL model, we do not use any spatial or temporal loss for ConvLSTM model but only the loss on labeled data and let the model find the spatial and temporal relationships automatically.

### Stacked FC-LSTM and CNN Encoder-Decoder interpolation

Besides the baseline is DAL model, we also implemented 2 more models based on Stacked FC-LSTM and CNN Encoder-Decoder to check how our proposed ConvLSTM better on both spatial and temporal features exploration. For Stacked FC-LSTM model (or FC-LSTM for short), we also use the input as the gray-scale image of 1024 pixel values. The time step input and output are the same as above models. We picked the number of hidden units for an LSTM cell is 2000 and stacked 3 LSTM cells to raise the model's capacity. The output of LSTM cells is then flowed through a fully connected neural network (FCNN) to produce the final output. Regarding CNN Encoder-Decoder model (or CNN for short), we applied an Encoder-Decoder network with the encoder is a convolution layer and decoder is a deconvolution layer similar to [Badrinarayanan]. To be comparable with ConvLSTM, we also used 1 encoder and 1 decoder layer with the same parameters as ConvLSTM (filter size is 3×3 and the number of output channels is 128). The RMSE results of CNN and LSTM model on the test set are shown in table 5. For additional comparison, we implemented ConvLSTM with the spatiotemporal loss as in DAL model and show the result in table 5.

Table 5: Compare RMSE results on the test set of 4 interpolation models

| Interpolation models | RMSE on test set |
|---|---|
| **ConvLSTM** | **8.31466** |
| DAL | 11.77393 |
| CNN Encoder-Decoder | 9.42967 |
| Stacked FC-LSTM | 12.01648 |
| **ConvLSTM + Spatiotemporal Loss** | **8.09817** |

From table 5, we can see that ConvLSTM model achieves the best RMSE among other baselines. Moreover, ConvLSTM model with spatiotemporal loss has better RMSE than pure ConvLSTM. It can be inferred that spatiotemporal loss is a good improvement for our addressing air pollution problem.

### Evaluations

The most critical evaluation for this part is to evaluate the citywide air pollution Interpolation. It means, how well the predicted output image reflects the air pollution of the whole city. In this part, we propose some techniques to evaluate this result.

Firstly, a model is better in the citywide interpolation if it can produce well air pollution values at existed monitoring stations' locations. That means the RMSE is small. It is easy to realize that the RMSE on the test set of proposed ConvLSTM is the best then following is CNN Encoder-Decoder and 2 last positions are DAL model and Stacked FC-LSTM.

The evaluation technique mentioned above is useful for quantitative evaluation but does not show us the overall picture of the interpolation result because the existing monitoring stations are sparse compared to the whole city. In figure 16, we plot the output images of DAL, ConvLSTM, CNN and FC-LSTM model to see the distribution of air pollution interpolation's values. Intuitively, the FC-LSTM model shows worst distribution output with all of the pixels except the existing monitoring stations have the same value because FC-LSTM network does not learn the spatial features well and thus does not give good interpolation output. For the remaining 3 models, ConvLSTM and DAL model show pretty good air pollution distribution compared to the CNN model. We propose some metrics to prove that ConvLSTM produces the better air pollution interpolation distribution compared to other baselines. To examine the goodness of interpolation distribution, we compare it with the actual air pollution values distribution. Here, we suggest using 2 metrics: the distribution variance and the Chi-squared test between distributions.

The first metric, variance, is the expectation of the squared deviation of a distribution from its mean. A high variance indicates that the data points are very spread out from the mean, and from one another. While a small variance indicates that the data points tend to be close to the mean and each other. For each evaluated model, we calculate the variance of actual air pollution values and the variance of interpolation values, do for 10 samples each and draw to the graph. From the graph in figure 17, we can see that the variance of interpolation distribution of ConvLSTM model is the closest to the variance of actual air pollution values distribution. That means ConvLSTM model outcomes better interpolation than DAL or CNN model.

The second metric, Chi-squared test, is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in a categorical variable. We can consider air pollution values are frequencies and check the chi-squared test between interpolated values and actual values. We also check for 10 samples of output for each model and compute the Chi-squared test between actual air pollution values distribution and interpolated values distribution. The results are shown in figure 18 and once again, ConvLSTM shows the smallest Chi-squared test against 2 remaining models, DAL and CNN Encoder-Decoder.

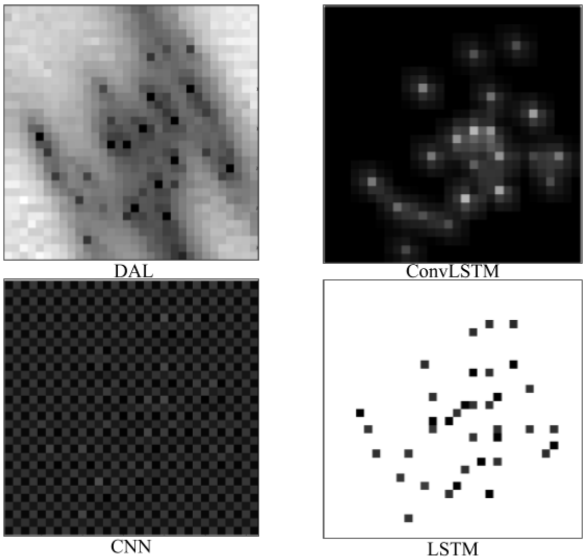


Figure 16: The plotting of interpolated output images of 4 interpolation models.

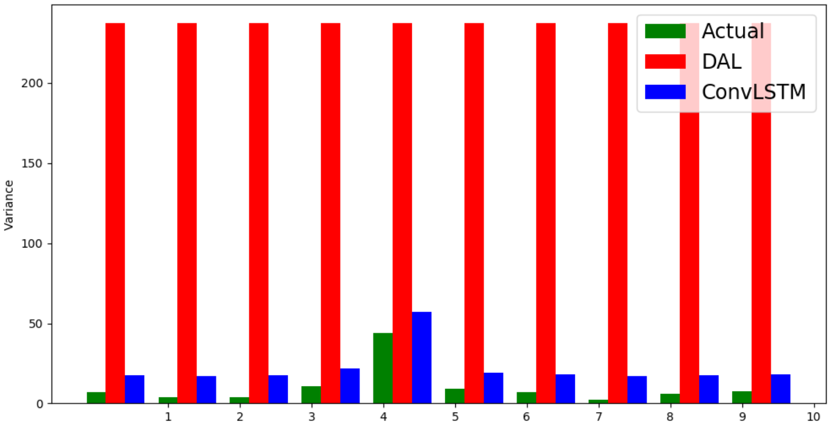


Figure 17: The Variance of actual values and interpolated values distribution of ConvLSTM and DAL model.

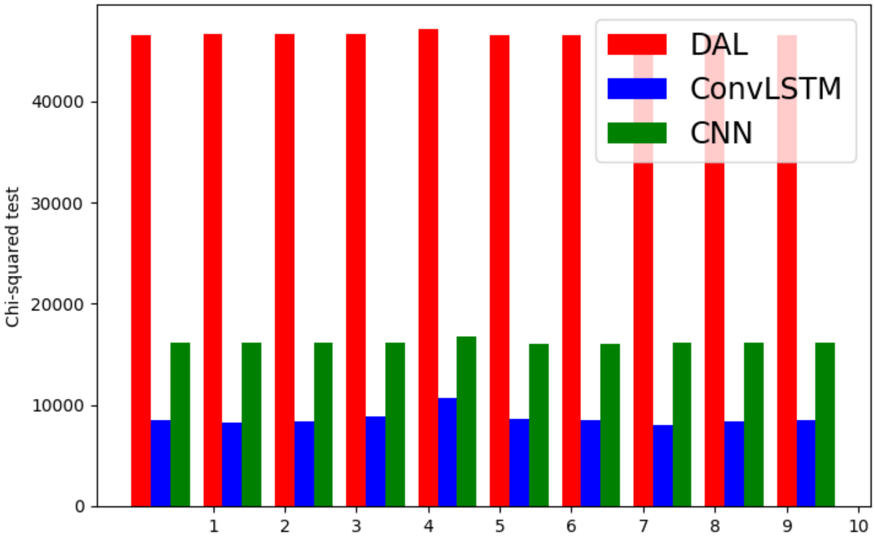


Figure 18: The Chi-squared test between interpolated and actual distribution of DAL, ConvLSTM, and CNN Encoder-Decoder model.

### Interpolation with air pollution influence factors

In this part, we show the experiment's results of air pollution interpolating with spatiotemporal air pollution impact factors like meteorology, transportation traffic and average speed, and outside air pollution sources. We did the experiments with following models: ConvLSTM (ConvLSTM with only air pollution data), ConvLSTM + Met (air pollution and meteorological data), ConvLSTM + Traffic (air pollution and transportation traffic data), ConvLSTM + Speed (air pollution and vehicles average speed data), ConvLSTM + Outside (air pollution and outside air pollution data), ConvLSTM + All (air pollution and all related factors). The RMSE results on the test set of pure ConvLSTM and other combination models are shown in table 6.

Table 6: Compare RMSE and spRMSE for air pollution interpolation of ConvLSTM network and its combination with other spatiotemporal factors

| Model | RMSE | spRMSE |
|---|---|---|
| ConvLSTM | 8.31466 | 15.48715 |
| **ConvLSTM + Met** | **6.58092** | 14.40496 |
| ConvLSTM + Traffic | 8.30858 | 15.47893 |
| ConvLSTM + Speed | 8.91373 | 15.17757 |
| ConvLSTM + Outside | 6.63926 | 14.46107 |
| **ConvLSTM + All** | 7.17028 | **11.02544** |

Following table 6, ConvLSTM + Met has the best RMSE which is intuitively reasonable because, in real, meteorology has the most significant impact to the air pollution. We also see that the RMSE of ConvLSTM + Speed model is not better than ConvLSTM. It can be explained that the average driving speed does not have significant fluctuations during the day.

To evaluate how other spatiotemporal factors impact the air pollution interpolation's efficiency we propose to use the following test: removing one of the existing air pollution values from input test data but still keep the values of other spatiotemporal data and then check the error of interpolated air pollution value with the existing one. If the error is small then we can infer that other spatiotemporal data has a remarkable effect on air pollution interpolation. To measure the error, we alternately set the air pollution value of each existing test input pixel to zero, keep other data unchanging, running the trained model on this modified input test data and calculate the RMSE between the inferred value with an actual same pixel value. The final error is the mean of all errors after doing this procedure with all test input pixels. We call this error spRMSE which means the RMSE caused by spatiotemporal factors. The experiment's results are shown in table 6. It can be seen that ConvLSTM + Speed model has a better spRMSE than ConvLSTM in spite of its worse RMSE which means the driving speed effects to air pollution in spatiotemporal form. ConvLSTM + All model has the best spRMSE which means we can improve the citywide interpolation with more spatiotemporal data.

### 4.2.2 Air pollution Forecasting: experiments and evaluations

#### Deep Air Learning (DAL) forecasting model

Firstly, we describe the baseline model for Air pollution forecasting, which is DAL forecasting model. DAL forecasting model has the same structure as DAL for interpolation but the input time steps are 24 hours before and the prediction time lags are 12 hours. We still pre-train an Auto-Encoder and then use it to train the prediction model. The spatial loss is computed by summing up the spatial loss for each 12 output image plates. The temporal loss is also the sum of the loss between 1 image slice with 2 neighbor image slices of the output (the DAL paper chose temporal neighbor size is 2). The final loss is the total of labeled loss and spatial and temporal loss.

### ConvLSTM forecasting model

The predicting ConvLSTM network also predicts 12 hours ahead from 12 previous hours as the input time steps. The number of encoder layers is 3 as the same number for forecasting layers. The output channels are 16, 16 and 32 respectively.

### CNN and FC-LSTM forecasting model

Similar to the interpolation experiment part, we also make 2 forecasting models based on CNN and FC-LSTM. The CNN model has 3 layers for encoder and 3 layers for the decoder part which is similar to ConvLSTM predicting model. To see how is the goodness of other spatiotemporal Deep Learning based models to our studying problem, we also implemented the LC-RNN model from [Zhongjian] paper which consists of some convolution layers following by a stacked LSTM. This model is also a combination of CNN and LSTM but in 2 continuous steps, not in 1 uniform model as in the ConvLSTM model.

Table 7 shows the RMSE of experimental models on the test set. As expected, ConvLSTM model gives the best RMSE, following is the CNN model and the last positions are DAL, LC-RNN and FC-LSTM.

Table 7: Compare RMSE of the forecasting models

| Interpolation model | RMSE on test set |
|---|---|
| **ConvLSTM** | **8.59883** |
| DAL | 9.44042 |
| CNN Encoder-Decoder | 9.16437 |
| Stacked FC-LSTM | 21.22256 |
| LC-RNN | 15.07063 |

### Forecasting with air pollution influence factors

Next, we conduct experiments with air pollution spatiotemporal related factors. The examining models are: ConvLSTM (as baseline), ConvLSTM + Met (air pollution and meteorological data), ConvLSTM + Traffic (air pollution and transportation traffic data), ConvLSTM + Speed (air pollution and vehicles average speed data), ConvLSTM + Outside (air pollution and outside air pollution data), ConvLSTM + All (air pollution and all related factors). Table 8 shows the RMSE of each examined models on the test set.

Table 8: RMSE of ConvLSTM model with spatiotemporal air pollution relating factors

| Model | RMSE |
|---|---|
| ConvLSTM | 8.59883 |
| **ConvLSTM + Met** | **8.43047** |
| ConvLSTM + Traffic | 8.53342 |
| ConvLSTM + Speed | 8.58124 |
| ConvLSTM + Outside | 8.53036 |
| ConvLSTM + All | 8.46117 |

Following table 8, the ConvLSTM + Met model has the best RMSE and ConvLSTM + All model takes the second position. Therefore, we still see the demonstration of affection of spatiotemporal factors into air pollution, especially by the meteorology.

For the last experiments' result, we produce the air pollution forecasting results of every hours from 1 to 12 hours. The results are shown in the graph in figure 19.

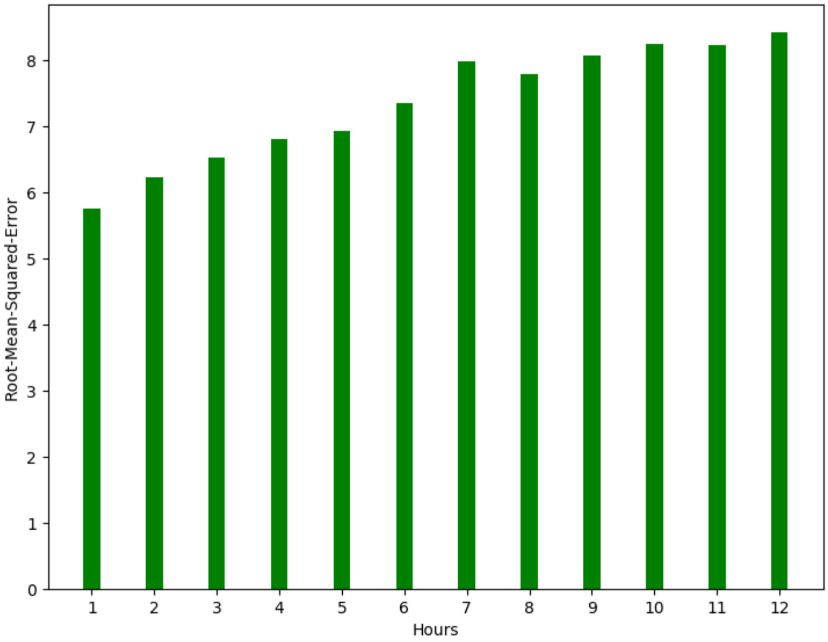


Figure 19: Air pollution forecasting results for every hours from 1 to 12 hours.

# Chapter 5

# CONCLUSIONS AND FUTURE WORK

In this chapter, we sum up all of our works so far on citywide Air pollution Interpolation and Prediction based on spatiotemporal Deep Learning models. We also evaluate our results and suggest the future extensions from this thesis's research.

## 5.1 Conclusions

To conclude, in this thesis, we have introduced 3 main contributions. Firstly, we have described and leveraged the citywide scale Air Pollution Interpolation and Prediction problem by considering a whole city to be one image. Secondly, we pointed out many spatiotemporal factors, which have affections to air pollution throughout the city. Lastly, we proposed a spatiotemporal Deep Learning based model for citywide air pollution interpolation and prediction. We have proved that the proposed ConvLSTM model does not only outperform state-of-the-art models but also works better than CNN and LSTM themselves in spatial and temporal features analysis. The combination of ConvLSTM and other spatiotemporal factors gives us a powerful model in interpolating and forecasting air pollution over the city.

## 5.2 Future work

In the future, we will try to improve the performance of the proposed model, both in the accuracy and the speed. Moreover, we can add more air pollution monitoring stations around the city or using real-time air pollution monitoring sensors installed on public transportation to better monitor air pollution for the whole city. The introduced solution is naturally fit to this new update through transfer learning with pre-trained models.

Our proposed ConvLSTM model for air pollution is also suitable for other urban spatiotemporal based predictions such as traffic volume prediction or crowd flow prediction. In the future, we will extend this spatiotemporal research on predicting urban traffic volume and driving speed to foresee traffic congestion and other urban relating problems.

# Bibliography

[Abadi] Abadi, Martín and et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.

[Beek] E.G. Beek, "Spatial interpolation of daily meteorological data," Wageningen (The Netherlands), DLO The Winand Staring Centre. Report 53.1, 1991.

[Cheng] W. Cheng, Y. Shen, Y. Zhu, L. Huang, "A Neural Attention Model for Urban Air Quality Inference: Learning the Weights of Monitoring Stations," *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

[Colah] C. Olah, "Understanding LSTM Networks," http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[EPA] United States Environmental Protection Agency, "Particulate Matter (PM) Basics," https://www.epa.gov/pm-pollution/particulate-matter-pm-basics#PM.

[Heo] J. Heo et al., "Two notable features in PM10 data and analysis of their causes," *Air Qual Atmos Health (2017)*, 10:991–998.

[Hsieh] H. Hsieh, S. Lin, Y. Zheng, "Inferring Air Quality for Station Location Recommendation Based on Urban Big Data," *KDD'15*, August 10 – 13, 2015, Sydney, Australia.

[Kampa] M. Kampa and E. Kastanas, "Human health effects of air pollution," *Environmental Pollution*, Volume 151, Issue 2, January 2008, pp. 362-367.

[Kriz] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *NIPS 2012*.

[Le] Van Duc Le, Sang Kyun Cha, "Realtime Air Pollution Prediction based on Spatiotemporal Big Data," *The International Conference on Big data, IoT, and Cloud Computing (BIC-18)*, August 20-22, 2018, Jeju, Korea.

[Li] L. Li et al., "Spatiotemporal Interpolation Methods for Air Pollution Exposure," *Symposium on Abstraction, Reformulation, and Approximation*, 2011.

[Mage] D. Mage et al., "URBAN AIR POLLUTION IN MEGACITIES OF THE WORLD," *Atmospheric Environment*, Vol. 30, No. 5, pp. 681-686, 1996.

[Mayer] H. Mayer, "Air pollution in cities," *Atmospheric Environment 33 (1999)*, 4029-4037.

[NIER-NASA] Korea's National Institute of Environmental Research (NIER) and the United States National Aeronautics and Space Administration (NASA), "The Korea-United States Air Quality Study (KORUS-AQ) report," 2016.

[Qi] Z. Qi et al., "Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[Shi] X. Shi et al., "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *NIPS 2015*.

[Wang] J. Wang et al., "Estimation of Citywide Air Pollution in Beijing," *PLoS ONE 2013*.

[Wiki] Wikipedia, "Air pollutants definition," https://en.wikipedia.org/wiki/Air_pollution.

[Wong] D. Wong et al., "Comparison of spatial interpolation methods for the estimation of air quality data," *Journal of Exposure Analysis and Environmental Epidemiology*, (2004) 14, 404–415.

[Zhang1] J. Zhang, D. Qi, “Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction,” *AAAI 2017*.

[Zhang2] Y. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, “DNN-Based Prediction Model for Spatio-Temporal Data,” *ACM SIGSPATIAL 2016*.

[Zhang3] Y. Zhang, W. Chan, N. Jaitly, “Very deep convolutional networks for end-to-end speech recognition,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2017*.

[Zheng1] Y. Zheng, F. Liu, H. Hsieh, “U-Air: When Urban Air Quality Inference Meets Big Data,” *KDD’13*, August 11–14, 2013, Chicago, Illinois, USA.

[Zheng2] Yu Zheng et.al., “Forecasting Fine-Grained Air Quality based on Big Data,” *KDD’15*, August 10 – 13, 2015, Sydney, Australia.

[Zhongjian] Zhongjian L. et al., “LC-RNN: A Deep Learning Model for Traffic Speed Prediction,” *JICAI*, 2018.

# 초록

대기 오염은 대도시에서 가장 큰 문제 중 하나이다. 많은 국가들은 주요 도시 주변에 대기 오염 모니터링 센터를 건설하여 대기 오염 물질을 수집하고 해당 지역의 시민들에게 대기 오염을 경고한다. 그러나 도시에서의 대기 오염은 균일하지 않으며 시공간 (spatiotemporal)적인 문제이다. 대기 오염은 위치 (공간적 특성)과 시각 (시간적 특성)에 따라 달라진다. 따라서, 도시 전체의 대기 오염 보간과 예측은 시민들이 시간과 공간에 대해 대기의 질을 파악하고, 나아가 건강에 대한 위협을 제거하기 위한 필요 조건이다. 대기 오염은 도시 전역의 여러 시공간적 요인에 의해 영향을 받는 것으로 알려져 있다. 그 중, 기상이 대기 오염에 가장 큰 영향을 주는 것으로 인식되고 있다. 그 외에, 교통량은 대기 오염의 주요 원인인 도로의 차량 밀도를 반영한다. 평균 주행 속도는 도시 대기 오염에 영향을 준다고 판단되는 교통 체증을 나타낸다. 마지막으로, 외부 대기 오염원은 도시 대기 오염 문제의 근원 중 하나라고 주장된다. 본 논문에서는 서울시의 대기 오염 데이터, 기상 데이터, 교통량, 평균 주행 속도와 같은 많은 시공간적 데이터와 서울의 대기 오염에 영향을 준다고 알려진 중국의 3개 지방(베이징, 상하이, 산동)의 대기 오염 데이터를 제시하였다.

대기 오염에 대한 최근의 연구에서는 특정 위치와 시간의 대기 오염 예측 모델을 구축하려고 시도해왔다. 그러나 대부분 연속되지 않은 위치에대한 대기 오염을 예측하거나 직접 만든 공간 및 시간적 특성을 사용하는 데 중점을 두었다. 최근 CNN (Convolutional Neural Network), RNN (Recurrent Neural Network) 및 LSTM (Long-Short Term Memory)과 같은 딥러닝 모델이 공간 및 시간 관련 문제에서 우수하다고 알려져있다. 본 논문에서는 CNN과 LSTM을 결합한 ConvLSTM (Convolutional

Long-Short Term Memory) 모델을 제안하였으며, 이를 통해 데이터의 공간 및 시간적 특성을 효율적으로 처리하고 최근의 다른 연구 결과보다 뛰어난 성능을 달성하였다.

**주요어**: 대기 오염, 보간, 예측, 도시 전체, 시공간, 딥러닝
**학번**: 2016-27885

# ACKNOWLEGEMENT

First of all, I would like to say thank you very much to Professor Cha Sang Kyun, who is also my advisor. Professor Cha is the founder and director of SNU Big Data Institute. He has guided me to choose this topic and helped me a lot in finding the data and building the model. He has inspired the research and entrepreneurship spirit to me and all members of our group.

Secondly, without the kind help from my colleagues in Peta-scale In-memory Database Laboratory, I cannot complete well this thesis. I also specially thank to Bui Tien Cuong, a Vietnamese student like me in the lab who supports me in the experiments doing and the thesis reviewing.

I also give my appreciation to all Professors of Department of Electrical and Computer Engineering, who equipped me with the advanced knowledge of Database, Computer Architecture, Programming Algorithms, and Machine Learning during my 2-year course in here.

Lastly, the great motivation behind me is my family (my wife and my little girl). They continuously encouraged and helped me during the hardest time of my Master's course. I would give all my love and my thankfulness to them.