M.S. THESIS

AQNet: Spatiotemporal Prediction of Air Quality Using Deep Generative Model

AQNet: 깊은 생성 모델을 이용한 대기 질의 시공간적 예측

BY

BUI TIEN CUONG

AUGUST 2019

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

M.S. THESIS

AQNet: Spatiotemporal Prediction of Air Quality Using Deep Generative Model

AQNet: 깊은 생성 모델을 이용한 대기 질의 시공간적 예측

BY

BUI TIEN CUONG

AUGUST 2019

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

AQNet: Spatiotemporal Prediction of Air Quality Using Deep Generative Model

AQNet: 깊은 생성 모델을 이용한 대기 질의 시공간적 예측

지도교수 차상균

이 논문을 공학석사 학위논문으로 제출함

2019년 8월

서울대학교 대학원

전기 컴퓨터 공학부

부이쿠옹티엔

부이쿠옹티엔의 공학석사 학위 논문을 인준함

2019년 8월

위 원 장: 정교민

부위원장: 차상균

위 원: 양인순

Abstract

With the increase of global economic activities and high energy demand, many countries have concerns about air pollution. However, air quality prediction is a challenging issue due to the complex interaction of many factors. In this thesis, we propose a deep generative model for spatio-temporal air quality prediction, entitled AQNet. Unlike previous work, our model transforms air quality index data into 2D frames (heat-map images) for effectively capturing spatial relations of air quality levels among different areas. It then combines the spatial representation with temporal features of critical factors such as meteorology and external air pollution sources. For prediction, the model first generates heat-map images of future air quality levels, then aggregates them into output values of corresponding areas. Based on the analyses of data, we also assessed the impacts of critical factors on air quality prediction. To evaluate the proposed method, we conducted experiments on two real-world air pollution datasets: Seoul dataset and China 1-year dataset. For Seoul dataset, our method showed a 15.2%, 8.2% improvement in mean absolute error score for long-term predictions of $PM_{2.5}$ and PM_{10}, respectively compared to baselines and state-of-the-art methods. Also, our method improved mean absolute error score of $PM_{2.5}$ predictions by 20% compared to the previous state-of-the-art results on China dataset.

keywords: Air Pollution Prediction, Deep Generative Models, Big Data Platform, Spatio-Temporal Prediction

student number: 2017-20722

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Air Pollution Problem

Air pollution is rapidly becoming a pressing issue for many large cities due to the rapid urbanization, along with concentrated economic activities. With the explosion of the global population and the rapid growth of world energy consumption, it eventually becomes a critical concern for many countries. These issues increase the need and demand for an accurate citywide air pollution prediction model, which is important for public health protection and government regulation. However, air quality (AQ) prediction is a challenging problem due to the involvement of multiple factors, including local pollutant emissions, coal power plants, dust activities, seasonal conditions, meteorology, terrain, and several other human activities according to [1][2]. Additionally, determining the correlation of critical factors with AQ levels is difficult due to their complex interactions. Besides, air quality relies not only on the temporal changes but also the spatial relations within the investigated area. Furthermore, the lack of homogeneous datasets to adequately assess the impact of critical factors on AQ levels is particularly apparent at the global scale.

1.2 Overview of the Proposed Method

Our approach is a novel data-driven for AQ prediction based on temporal and spatial properties of heterogeneous data sources. As mentioned earlier, local air quality is governed by many critical factors. According to the WHO report in 2006 [3], $PM_{2.5}$ (particulate matter has diameter $\leq 2.5\mu m$) can travel from hundreds to thousands of kilometers and remain suspended in the air for weeks. However, PM_{10} (particulate matter has diameter $\leq 10\mu m$) can only disperse up to a few hundred kilometers and persist for a period ranging from a few minutes to several days. Based on these findings, we collected observational data (air pollutant concentrations, meteorology), weather forecasts, and categorical features (time, date) of investigated cities. Time and date information includes holidays, hours in a day, and months in a year which indirectly reflect effects of transportation density or seasonal weather conditions. The air pollutant concentrations (AQ data) comprise gaseous pollutants (NO_2, SO_2, CO, and O_3) and particulate matters ($PM_{2.5}$ and PM_{10}). Then, our model predicts two of the most crucial factors, $PM_{2.5}$ and PM_{10}, which have chronic effects on human health according to [4] [5].

In this thesis, we develop a deep generative model for spatio-temporal air quality prediction, entitled AQNet. The deep generative model consists of two essential factors: a context vector and a generative process. First, a spatial transformation function converts the local AQ data into heat-map images to analyze spatial changes of air quality. By using images, our model effectively captures spatial properties and relationships without explicitly modeling the spatio-temporal dynamics of $PM_{2.5}$ or PM_{10} and other atmospheric factors. Next, it extracts valuable features from the heat-map images by exploiting the spatial learning processes of MS-CNN layers. Secondly, the model combines the spatial representation with temporal features of critical factors into an encoded vector. This vector is then concatenated with the representation output of a previous prediction image to provide a context vector for the next step of the generative process. By generating images, the model is forced to focus on spatial relations among areas and predict better for long-term AQ levels. Additionally, the deep generative model can combine information of various data sources with an increase in the accuracy of predictions, while putting additional data sources to other baselines and state-of-the-art methods degrades their correctness.

To prove the efficacy of the proposed method, we performed experiments on real-world air pollution datasets. Experiments were first conducted on Seoul dataset, and then the model robustness was evaluated using China 1-year dataset. For Seoul dataset, we compared our model's results with baselines and state-of-the-art methods in hourly (up to 24 hours) and long-term (e.g., 1^{st} day, 2^{nd} day) predictions. Our method showed 15.2%, 8.2% improvement of the mean absolute error for long-term predictions of $PM_{2.5}$ and PM_{10} compared with other methods on Seoul dataset. The small size of China dataset is a big obstacle for deep learning models. To overcome this issue, we applied a transfer learning method that uses the pre-trained weights of Seoul dataset for China dataset. As a result, our method improved the mean absolute error of $PM_{2.5}$ predictions by 20% compared to the previous method. We also conducted an experiment to train our model on China data from scratch, and the results were 11% better than the previous method on the mean absolute error.

Based on real-world datasets, we assessed the impact of critical factors on AQ prediction. We conducted various experiments on all combinations of data sources, including local AQ data, air pollution data of neighboring countries, and the meteorological data. For instance, to assess the impact of weather conditions on AQ levels, the model was trained to predict future AQ levels using only meteorological data. The experimental results revealed that the local $PM_{2.5}$ levels directly rely on the meteorological conditions and the $PM_{2.5}$ variations of neighboring countries, while PM_{10} levels are more localized and less sensitive to these factors.

1.3 Contributions

Our contributions are summarized as follows:

- We propose a deep generative model for spatio-temporal air quality prediction, entitled AQNet. The model considers both spatial relations of air quality levels among areas and temporal features of various critical factors.

- Our model transforms AQ data into heat-map images for efficiently capturing spatial changes of AQ levels among areas. To predict future AQ levels, the model first generates heat-map images, then aggregates pixels into output values for corresponding areas.

- Based on the analyses of data, we assess the impacts of critical factors on the air quality prediction.

- On Seoul dataset, our method has 15.2%, 8.2% improvement of the mean absolute error on long-term predictions of $PM_{2.5}$ and PM_{10} compared to previous baselines and state-of-the-art methods. On China 1-year dataset, we improve the mean absolute error of $PM_{2.5}$ predictions 20% compared to the previous work.

Chapter 2

RELATED WORK

2.1 Spatio-Temporal Prediction

The spatio-temporal prediction has been prevalent in many domains, especially urban problems such as air pollution, traffic congestion, weather nowcasting, crowd flow prediction, to name a few. In terms of precipitation nowcasting, Shi et al., 2015 [6] proposed a model that updates traditional FC-LSTM by replacing matrix multiplication with convolution operation to predict future precipitation based on historical precipitation radar images. Donahue et al., 2015 [7] proposed LRCNs, a class of architectures for visual recognition and description, which combined convolutional layers and long-range temporal (CNN-LSTM). Our work is similar to LRCNs in terms of the order of learning processes, in that the model first captures spatial relations, then feeds spatial representation output to the temporal module. In traffic domain, Yu et al., 2017 [8] and Li et al., 2017 [9] proposed spatio-temporal models for speed forecasting. Yao et al., 2018 [10] proposed a CNN-LSTM model for taxi demand prediction. In crowd flow prediction, Zhang et al. [11] proposed a deep spatio-temporal method based on convolutional residual networks. Among these methods, CNNs were widely used for learning and tracking spatial representation, while LSTMs were usually used for processing time series sequence. LSTMs were also the backbone of numerous time-series problems in many domains such as natural language processing [12] [13], stock prediction [14], to name a few. In air pollution research, not many methods applied original CNNs due to the spatial sparsity of monitoring data. Some works took into account exact locations of stations and applied derivative versions of CNNs such as graph convolutional networks [15] or diffusion convolution networks [16] for spatio-temporal prediction. In this thesis, we propose an efficient way of transforming local AQ data into heat-map images and make use of CNN-LSTMs for spatio-temporal air pollution prediction.

2.2 Air Pollution

Air pollution problem has received considerable attention from numerous researchers across diverse subjects. In the area of environmental science, researchers tried to identify the root cause of air pollution and the correlation with critical factors. These researches approached classical methods such as Gaussian models [17] [18] or SVR [19], but they were difficult to apply to real-time forecasting systems. The data deficiency of various sources, including meteorology, transportation, industrial activities, and AQ data, made this problem even more challenging for traditional methods. Therefore, these methods were difficult to provide reliable forecasts, especially in real-time. We instead approached a data-driven method to predict the two most concerned objects ($PM_{2.5}$ and PM_{10}).

Data-driven models aggregate numerous data sources to provide accurate prediction results. The advent of deep learning and big data facilitates the ubiquity of data-driven approaches for AQ prediction. Zheng et al. proposed [20] and [21], which were the very first research applying big data techniques for AQ inference. However, the methods proposed in these papers were also based on basic machine learning methods such as feature extraction and ensemble models. Besides that, spatio-temporal deep learning models for AQ prediction focused on predicting AQ levels of monitoring stations based on weighted networks [22] [23]. These works are similar to us in terms of feature embedding for meteorological conditions, time encoding, and AQ data. However, they did not assess the exact influence of critical factors on short-to-long-term AQ levels. Our approach addresses this problem and predicts short-to-long-term AQ levels. It integrates the spatio-temporal features of heterogeneous data sources to not only give superior results but also assess the impacts of critical factors on AQ levels.

Chapter 3

OVERVIEW

3.1 System Architecture

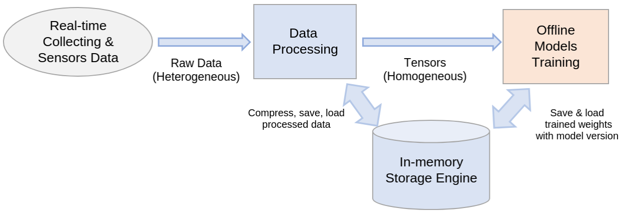

Real-time Collecting & Sensors Data
Raw Data (Heterogeneous)
Data Processing
Tensors (Homogeneous)
Offline Models Training
Compress, save, load processed data
In-memory Storage Engine
Save & load trained weights with model version

Figure 3.1: Offline learning system. We train all data combinations and manually select corresponding components.

Our system consists of two modes, offline learning and online learning sharing multiple components. As depicted in Fig. 3.1, offline learning mode comprises four components, including the real-time data processing, model training, in-memory storage engine, and a front-end application. We can implement and update components separately since each one of them is independent of others. Fig. 3.2 shows that the online learning structure is only different from the offline mode in the model execution component, the gradient method, and the visualization method. We elaborate our prediction model in Chapter 5, and the user interfaces in Section 3.2. In this session, we focus on explaining data sources and the system flow.

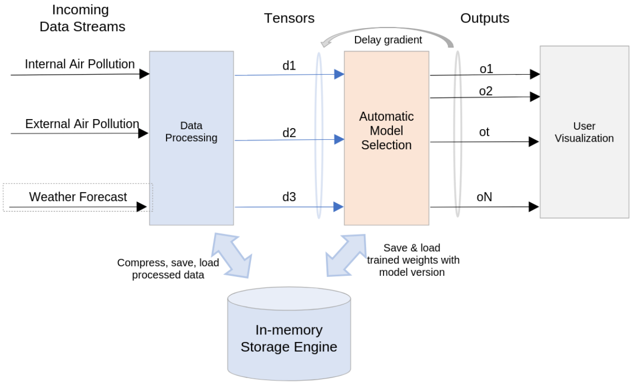


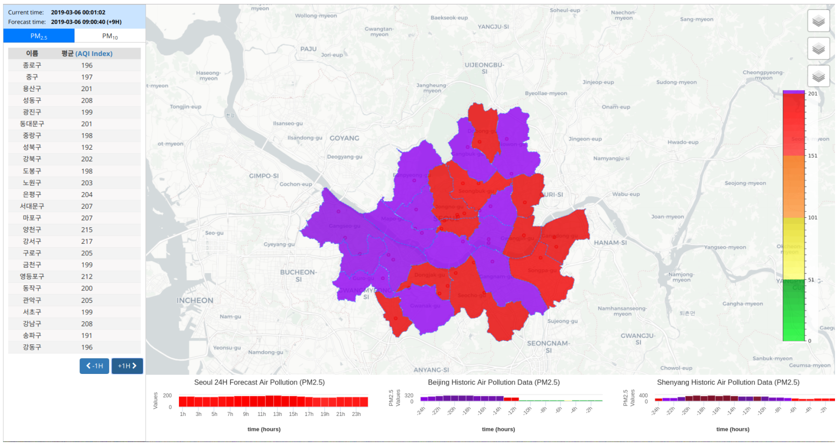
Figure 3.2: Online learning and real-time prediction system. The system selects the execution model corresponding to data availability.

First, the system processes data from multiple sources into tensors, then feed them to designed models for training and testing. All models are implemented in Tensorflow. In the offline phase, the models are trained for a few hours and saved the best-learned weights. These weights are reloaded later during the online prediction period. The predictive system generates results for both particulate matter $PM_{2.5}$ and PM_{10}. A public web server is developed in CherryPy for receiving RESTful requests from clients via a web application. After receiving a request, the system activates a real-time data processing component, then sends an execution signal to the predictive model. Finally, prediction results are sent back to the client in JSON format for displaying.

Next, In-memory Storage Engine, an essential component of our platform, provides compression, loading, storing, and computing operators for data processing, models management. Air Quality Forecasting system includes heterogeneous data streams demanding a great computing potential. Furthermore, time series data streams, including million records collecting every millisecond cause big troubles for conventional storage systems. Besides that, to provide reliable forecasts in online learning mode, the system needs to automatically select the most appropriate and accurate model depending on currently available data. The system is trained with numerous models based on multiple data combinations. Fig. 3.2 presents three data streams that create seven possible data combinations corresponding to seven different models. However, it raises concerns for pre-trained weights and models management. In-memory storage engine provides powerful tools to satisfy these mentioned demands.

3.2 User Interface

To visualize real-time air quality forecasts, we create a web application based on AngularJS 4 and LeafletJs as depicted in Fig. 3.3. The application shows the prediction map of 25 districts in Seoul, hourly (up to 24h) and daily (up to 7 days) averaged values of the city for two most concerned AQIs $PM_{2.5}$ and PM_{10}. The left box shows the selected forecast of 25 districts of the city navigated via two blue buttons at the bottom. Besides that, it also presents historical air quality variations of neighbor areas. The color bar uses in Fig. 3.3 based on standard AQI scale as shown in the bottom right corner. The system automatically sends updating a Restful request to get the latest forecasts every one hour or at the loading time.



Current time: 2019-03-06 00:01:02
Forecast time: 2019-03-06 09:00:40 (+9H)

이름	평균 (AQI Index)
종로구	196
중구	197
용산구	201
성동구	208
광진구	199
동대문구	201
중랑구	198
성북구	192
강북구	202
도봉구	198
노원구	203
은평구	204
서대문구	207
마포구	207
양천구	215
강서구	217
구로구	205
금천구	199
영등포구	212
동작구	200
관악구	205
서초구	199
강남구	208
송파구	191
강동구	196

PM2.5
PM10
< -1H
+1H >
PAJU
GOYANG
UIJEONGBU-SI
GIMPO-SI
BUCHEON-SI
INCHEON
HANAM-SI
SEONGNAM-SI
GWANGJU-SI
ANYANG-SI
201
151
101
51
0
Seoul 24H Forecast Air Pollution (PM2.5)
Values
200
0
1h 3h 5h 7h 9h 11h 13h 15h 17h 19h 21h 23h
time (hours)
Beijing Historic Air Pollution Data (PM2.5)
PM2.5 Values
320
0
-24h -22h -20h -18h -16h -14h -12h -10h -8h -6h -4h -2h
time (hours)
Shenyang Historic Air Pollution Data (PM2.5)
PM2.5 Values
400
0
-24h -22h -20h -18h -16h -14h -12h -10h -8h -6h -4h -2h
time (hours)

Figure 3.3: Our application for real-time air quality prediction. We instantly access heterogeneous data sources and provide real-time predictions.

Chapter 4

DATA MANAGEMENT

4.1 Real-time Data Collecting

The biggest obstacle to air pollution research is a lack of reliable data resources. Even though some organizations publish historical records of particulate matters, they focus on only ($PM_{2.5}$) while lacking information about other gas pollutants. Meteorology is also a significant impact on air pollution in multiple urban cities, but it is not easy to collect reliable historical data. We implement a real-time system that can instantly collect periodic meteorological data (1h, 3h), hourly air pollution information including standard pollutants of Seoul, and particulate matter 2.5 information of Beijing, Shandong, Shenyang. Our collected data for Seoul city consists of specific information of 25 districts. Elaborately, we collect meteorology data of two websites, including World Weather Online and Seoul autonomous weather system (aws). We gather China particulate matter 2.5 data from US Embassy portal and Berkeley Earth website, and the Seoul air pollutant data from the government CleanAir and AirKorea website. All collected data are ten years in length from 2008 to 2018 except the Shandong $PM1_{2.5}$ which comes from Berkeley earth and is only from May 2014. We implement the real-time data collecting system in Python 2.7.

4.2 Data Collection

Seoul Air Pollution Dataset. This dataset was collected from multiple public sources of the Seoul Government and consisted of real-time data of 25 districts averaged over 39 measurement stations across the city from 2008. AQ data include hourly observations of gaseous pollutants (NO_2, SO_2, CO_2, O_3) and particulate matters ($PM_{2.5}$, PM_{10}). We then normalized input vectors using the min-max scale. To reduce the execution time, we pre-processed AQ data of an investigated area into heat-map images using a spatial transformation function, which is addressed in Section 5.3. We used data only from 2014 to 2018 for training, testing, and validating. We trained our model from 2014 to 2016 and used 2017, 2018 data as the validation set and test set.

Neighboring Areas Air Pollution Data. The air quality variations of Seoul are the results of both local and external air pollution sources. In the scope of this research, we consider the air pollution sources of some cities in China (Beijing, Shandong, Shenyang) as an external factor, which affects Seoul AQ levels. We only used $PM_{2.5}$ data of these cities published in the US Embassy website and Berkeley Earth website as the representation of their air pollution levels due to the lack of available public sources. All data were shifted one hour (+1h) to match the timestamp of the Seoul dataset (Seoul GMT +9).

Meteorological Data. We consider meteorology as a critical factor for future AQ levels and collected a long period meteorological data in Seoul and neighboring cities from two different sources. First, we crawled weather forecast data of all cities (Seoul, Beijing, Shenyang, and Shandong) from WorldWeatherOnline.com. Second, we obtained meteorological observational data of all areas in Seoul from the Seoul Government's autonomous weather system website. Both data sources include standard meteorological information such as wind speed, wind direction, humidity, temperature, and precipitation. The range of data in each city was different due to the availability of publishing sources. Therefore, Seoul's data were crawled from 2008, while the other cities' data were collected from 2009. Next, we matched the timestamp of meteorological data with other data sources.

Additional Information. Air pollution is also associated with many other factors, including local transportation and seasonal features, for which data are either unavailable or difficult to obtain, especially from Chinese sources. Therefore, we added categorical features such as months in a year, hours in a day, and a binary holiday flag to input vectors. For example, the month-in-year feature can indirectly show the seasonal features, while hour-in-day element and the holiday flag can represent local traffic volume at specific times.

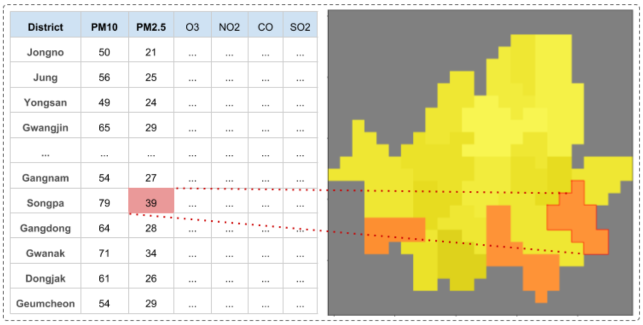
China 1-year dataset. This dataset was published in Zheng et al., 2015 [24], named as FFA (Forecasting Fine-grained Air pollution) and collected from May 2014 to April 2015. It consists of AQ data, meteorological data, and weather forecasts of many Chinese cities. These data are similar to Seoul air pollution and meteorological data, as mentioned earlier. However, the length of this dataset is only one year, which causes difficulties for deep learning approaches. Besides, a large amount of data is missing in it. For instance, PM_{10} has 38% missing records out of 278,023 records for Beijing.

4.3 Spatial Transformation Function

Our datasets have two different structures, in which one contains positional information (latitude and longitude) of measurement stations while the other consists of only districts aggregation values. Datasets, which include positional information such as AirKorea or China 1-year data, are convenient and straightforward for spatial and temporal interpolation. However, these datasets are usually neither instantly updated nor available for crawling, causing difficulty for real-time interpolation and prediction. On the other hand, the AQ pollutant data of Seoul Government (CleanAir Website) comprise hourly data of 25 districts in Seoul from 2008.

The spatial transformation function converts the local AQ data into heat-map images to analyze spatial changes of air quality. By using images, our model effectively captures spatial properties and relationships without explicitly modeling the spatio-temporal dynamics of $\mathrm{PM}_{2.5}$ or PM_{10} and other atmospheric factors. Next, it extracts valuable features from the heat-map images by exploiting the spatial learning processes of CNN layers. Besides that, using images allow the predictive model to interpolate missing values spatially. As a result, the original problem is converted to predicting future heat-map images from sequences of historical ones. From practical reasons, the size of an image is set as 25x25 or 32x32 for computation efficiency and prediction accuracy. We propose two different spatial interpolation methods for each type of dataset as follows.

4.3.1 District-based Interpolation



District	PM10	PM2.5	O3	NO2	CO	SO2
Jongno	50	21	...	...	...	...
Jung	56	25	...	...	...	...
Yongsan	49	24	...	...	...	...
Gwangjin	65	29	...	...	...	...
...	...	...	...	...	...	...
Gangnam	54	27	...	...	...	...
Songpa	79	39	...	...	...	...
Gangdong	64	28	...	...	...	...
Gwanak	71	34	...	...	...	...
Dongjak	61	26	...	...	...	...
Geumcheon	54	29	...	...	...	...

Figure 4.1: Spatial transformation method. It transforms row-wise data into heat-map images. Each area's pixels have the same value as its concentration value.

Air pollutant data are crawled from CleanAir website, including NO_2, CO, SO_2, O_3, $PM_{2.5}$, PM_{10} of 25 districts in Seoul. This dataset does not include geometrical information such as latitude and longitude of measuring stations, which causes difficulties to the citywide level interpolation. We assume that every location in a district has a level of air pollution as same as the representative value. Therefore, we create a heat-map map image for each hour data of Seoul where all pixels of a district hold an equivalent number that is the instant crawled value as presented in Fig. 4.1. We firstly specify the boundary of each district in a heat-map image then fill all pixels in this area with its data. We repeat this process for each air pollutant mentioned above. As a result, each hour data of districts is converted to a cube with size Width x Height x Dimension where width and height are corresponding to the image while dimension depends on the number of pollutants.

4.3.2 Geo-based Interpolation

Geo-based interpolation method also generates a heat-map image from hourly data, but it is a little different from the previous method. First, we convert the latitude and longitude of each station into coordinate values as follows:

$$
\begin{aligned}
x &= \lfloor \frac{lng - a_x}{b_x} \rfloor \\
y &= \lfloor \frac{lat - a_y}{b_y} \rfloor \\
idx &= y * s + x
\end{aligned}
\tag{4.1}
$$

where lat and lng are the latitude and longitude value of a station. The longitude value of the left vertical boundary and the latitude value of the bottom boundary line are denoted as a_y and a_x, respectively. The width and the height of a grid cell are b_x and b_y, respectively. Let s denote the size of a heat-map image. Finally, idx is the index of a station in the 1-D flattened array of a heat-map image.

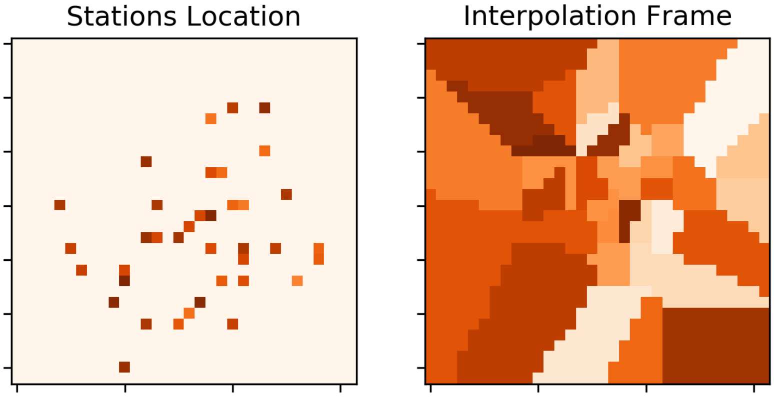


Figure 4.2: Transformation of row-wise values of stations to a heat-map image.

We fill stations' idx with their corresponding values then apply the k-nearest neighbors (k-NN) algorithm to interpolate the remaining points. Fig. 4.2 presents an example of a geo-based interpolation heat-map image. This method allows us to interpolate the missing points efficiently spatially. Up to our knowledge, not many data sources contain geo-information and also allow instant access, which is an obstacle to this interpolation method for applying to a real-time predictive system. Therefore, we only use this method for experimental purposes while sticking with the approach, as described in 4.3.1, for the real-time prediction system.

Chapter 5

Proposed Method

5.1 Data Source

First, we define data sources used in our model as follows:

- $\mathcal{I} = \{I_t^k\}$: AQ data of an **I**nvestigated area
- $\mathcal{M} = \{M_t^k, W_t^x\}$: **M**eteorological data
- $\mathcal{N} = \{N_t^x\}$: AQ data of **N**eighboring countries
- $\mathcal{D} = \{D_t^x\}$: Additional **D**ate information

For brevity, index t, k, and x represent each time step, each station, and each city, respectively. Also, x can be either s or c, which represent the investigated area or cities from neighboring countries, respectively. I_t^k is the 6-dimensional vector of AQI concentrations in station k for time step t. Similarly, N_t^x denotes the 1-dimensional vector of $\text{PM}_{2.5}$ concentrations of a city x for time step t. M_t^k and W_t^x mean the 6-dimensional vector of meteorological conditions, including temperatures, humidity, precipitation, wind, and weather forecasts. Additionally, M_t^k denotes the meteorological observational data, while W_t^x is the weather forecasts of a city x. Lastly, D_t^x denotes the

3-dimensional vector of date information of a city x, including a binary holiday flag, months, and hours.

Table 5.1: Input Vector Features

Data Source	Feature	Range	Type
$\mathcal{I}$	$PM_{2.5}$	[0,1]	Float
	PM_{10}	[0,1]	
	O_3	[0,1]	
	NO_2	[0,1]	
	SO_2	[0,1]	
	CO	[0,1]	
$\mathcal{M}$	Temperature	[0,1]	Float
	Humidity	[0,1]	
	Precipitation	[0,1]	
	Wind Speed	[0,1]	
	Wind Gust	[0,1]	
	Wind Direction	[0,360]	Categorical
$\mathcal{N}$	$PM_{2.5}$	[0, 1]	Float
$\mathcal{D}$	Month	[1,12]	Categorical
	IsHoliday	$\{0,1\}$	
	Hour	[0,23]	

Based on the statistical data analysis, we specified the input features of data sources as described in Table 5.1. The features of $\mathcal{I}$ and $\mathcal{M}$ consist of all AQI and standard meteorological conditions. Unlike $\mathcal{I}$, $\mathcal{N}$ has only $PM_{2.5}$ due to the lack of AQ data from open sources from neighboring countries. Also, we added to vectors $\mathcal{D}$'s features corresponding to each country to represent the date and time properties.

We categorized input features into two types, namely categorical features and numerical features. For categorical features, we used numeric encoding instead of one-hot encoding, since the sparseness of one-hot encoding causes expensive computation and degrades the model performance. For numerical features, we normalized them using the min-max scaling method.

5.2 Problem Definition

Given $\mathcal{I}, \mathcal{M}, \mathcal{N}, \mathcal{D}$, the model f_θ aims to predict $\widehat{Y}$ which is the air quality (PM$_{2.5}$ and PM$_{10}$ concentrations) over the next T_{pred} hours for each station $k \in K$.

$$f_\theta\Big(\mathcal{I}, \mathcal{M}, \mathcal{N}, \mathcal{D}\Big) = \widehat{Y} \tag{5.1}$$

Our goal is to train the model f_θ using collected datasets with L_2 loss functions.

5.3 Model

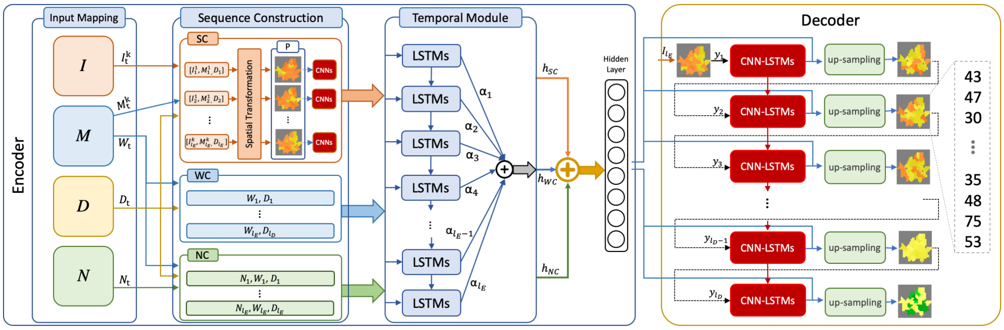


Figure 5.1: An overview of AQNet. It is based on the encoder-decoder framework with CNN-LSTM networks. The encoder consists of three main modules, which take input vectors of four data sources into account and output a 128-dimensional vector. The decoder generates heat-map images of future air quality based on CNN-LSTM units and up-sampling networks.

To capture features from various data sources, we designed a deep generative model based on an encoder-decoder framework. As depicted in Fig. 5.1, our model comprises of two main components, an encoder and a decoder.

5.3.1 Encoder

The encoder comprises an input mapping module, a sequence construction module, a temporal module, and a hidden layer feedforward network.

Input Mapping. This module consists of normalized vectors of four data sources $\mathcal{I}, \mathcal{M}, \mathcal{N}, \mathcal{D}$. As Eq. 5.2 shows, it concatenates vectors of different data sources into input vectors for each part in the sequence construction module. We denote $\mathcal{SC}$, $\mathcal{WC}$, and $\mathcal{NC}$ as the input sets of **SC**, **WC**, and **NC**.

$$\begin{aligned} \mathcal{SC} &= \text{Concatenate}\big(\mathcal{I}, \mathcal{M}, \mathcal{D}\big) \\ \mathcal{WC} &= \text{Concatenate}\big(\mathcal{M}(W^s), \mathcal{D}\big) \\ \mathcal{NC} &= \text{Concatenate}\big(\mathcal{N}, \mathcal{M}(W^c), \mathcal{D}\big) \end{aligned} \tag{5.2}$$

Sequence Construction. The sequence construction module comprises of three parts for **SC**, **WC**, and **NC**, which aim to construct sequences of input vectors from data sources for the temporal module. It is important to note that **SC** is used to capture spatial relations of observational data, while **MC** and **NC** are employed to capture temporal changes of future meteorology (weather forecasts) and air pollution sources from neighboring countries. The roles of each component can be described as follows:

- **SC**'s inputs are the combination of AQ data, meteorological observational data, and additional date information of the investigated area. Each vector comprises of 15 features of $\mathcal{I}$, $\mathcal{M}$, and $\mathcal{D}$ in Table 5.1. **SC** has a spatial transformation function, which turns input vectors into heat-map images to capture spatial relations of air quality and meteorology among various areas. Finally, it gives an output of a sequence of 128-dimensional vectors.

- **WC** captures the temporal changes of future meteorological conditions of the investigated area to assess their impact on the variation of future air quality. **WC**'s inputs are 9-dimensional vectors, which comprise of features of $\mathcal{M}$ and $\mathcal{D}$. D's features represent additional information such as seasonal variations through time and date information. Next, it loads a sequence of vectors and pushes them to the temporal module.

- **NC** is responsible for determining the impact of transboundary air pollution sources on local AQ levels. Its inputs are 27-dimensional vectors, which comprise of 21 (7x3) features of $\mathcal{N}$ and $\mathcal{M}$ of three cities in China, and 3 features of $\mathcal{D}$. Particularly, $\mathcal{D}$'s features allow our model to represent additional information such as seasonal variations, transportation situation, to name a few. These features are critical to revealing the variations of external air quality since $\mathcal{N}$ has only $PM_{2.5}$ data.

Spatial Transformation. The spatial sparsity of observational data of monitoring stations causes difficulties in capturing the variations of air quality in areas. To take into account additional spatial information in areas, we used heat-map images instead of row-wise data for training. In the **SC** module, we designed a spatial transformation function to convert row-wise data into heat-map images of 25×25 size. First, we developed a tool to define the boundary of districts in the image. Then, the spatial transformation function filled all pixels of a district in an image with a corresponding AQI value as depicted in Fig. 4.1. Algorithm 1 depicts a group of heat-map images as $\mathcal{P}$. The inputs of **SC** turn to sequences of heat-map images instead of input vectors for districts. Next, the model pushes these images to a convolutional-based network for capturing spatial relations. To enable the model to learn both local and global spatial features of the images, we designed a multi-scale convolutional neural network (MS-CNN) model similar to the naive version of [25]. Table 5.2 demonstrates the structure of MS-CNN of the **SC** component. The **SC**'s output is now a sequence of vectors similar to the other parts in the sequence construction module.

Table 5.2: Configurations of MS-CNNs for AQNet Encoder

Layers	Type	Num.	Kernel Size	Output Size
Down-sample	ReLU	32	5×5	11×11×32
Multi-scale Feature	MS-CNN ReLU	4×16 -	(7,5,3,1)×(7,5,3,1) -	11×11×64 -
Down-sample	ReLU	32	5×5	4×4×32
Multi-scale Feature	MS-CNN ReLU	4×8 -	(7,5,3,1)×(7,5,3,1) -	4×4×32 -
Down-sample	ReLU	32	5×5	2×2×32
Output	Flatten	-	-	128

As shown in Fig. 5.2, multi-scale convolutional networks, used in our model, are comprised of multiple filters with different kernel size (including 7x7, 5x5, 3x3, 1x1) as similar to the naïve version of inception module [26]. The size of kernels is decided based on intuitive reasons rather than specific concepts. Each sub-convolution networks work as similar to the original version which applies kernel function to local regions then concatenates the outputs along the last dimension into a single one forming the next stage's input. Multi-scale networks can efficiently learn both local and global features of input through multiple kernels with different size.

Temporal module. The temporal module, which is based on attention-based networks with LSTM units, is responsible for outputting a 128-dimensional hidden vector from a sequence of vectors in each part of the sequence construction module. The model uses LSTM to learn the temporal changes of critical factors, which affect future AQ levels. To learn the varying importance of each time step, we included a weighted attention layer on top of LSTM outputs. The module initializes different weights for **SC**, **WC**, and **NC**.

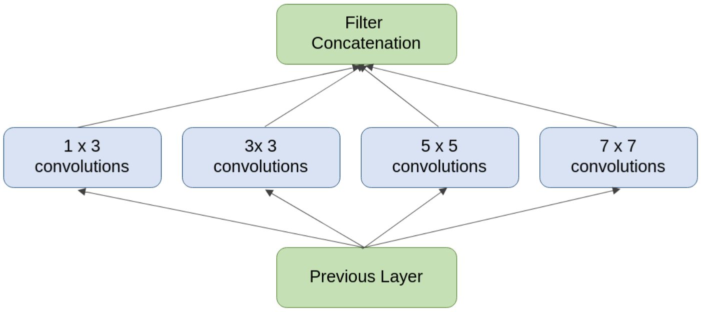


Figure 5.2: A Multi-scale Convolutional Unit with different kernels size.

Long short-term memory network (LSTM) is a variance of a recurrent neural network (RNN), which are prevalent in time series analysis. Recurrent Neural Networks consist of a sequence of inputs $\mathbf{x} = (x_1, ..., x_\mathrm{T})$ and activation units. In each time step, the activation unit computes the hidden vector h and the output vector y. After T time steps, the networks generate a sequence of hidden and output vectors. The networks are described as following:

$$\begin{aligned} h_\mathrm{t} &= \sigma\left(W_\mathrm{h} x_\mathrm{t} + U_\mathrm{h} h_\mathrm{t\text{-}1} + b_\mathrm{h}\right) \\ y_\mathrm{t} &= \sigma\left(W_\mathrm{y} h_\mathrm{t} + b_\mathrm{y}\right) \end{aligned} \tag{5.3}$$

where the term W and U denote weights (e.g., W_xh) is the weight of the hidden layer corresponding to input vector x, the term b denotes the bias vector (e.g., b_h is the bias of the hidden layer), and σ is the activation function such as Sigmoid, Tanh, or ReLU.

Conventional RNNs confront with either vanishing or exploding gradients problem causing poor performance when executing long sequences. Long Short-term Memory and other derivative architectures solve these problems by replacing the existing activation function with a robust network what can both learn and remember knowledge. They all have specific memory cells with complicated dynamics allowing them to memorize trained knowledge much better than the original RNNs. Because of these properties, a modern LSTM architecture can easily extend and stack on top of others to form multiple layers structure. Practically, we apply a simple version of LSTM networks with only one layer for better execution time while still producing noticeable results. The LSTM architecture, used in our models, is given by the following equations:

$$
\begin{aligned}
i_{\mathrm{t}} &= \sigma\big(W_{\mathrm{i}}x_{\mathrm{t}} + U_{\mathrm{i}}h_{\mathrm{t\text{-}1}} + b_{\mathrm{i}}\big) \\
f_{\mathrm{t}} &= \sigma\big(W_{\mathrm{f}}x_{\mathrm{t}} + U_{\mathrm{f}}h_{\mathrm{t\text{-}1}} + b_{\mathrm{f}}\big) \\
o_{\mathrm{t}} &= \sigma\big(W_{\mathrm{o}}x_{\mathrm{t}} + U_{\mathrm{o}}h_{\mathrm{t\text{-}1}} + b_{\mathrm{o}}\big) \\
c_{\mathrm{t}} &= f_{\mathrm{t}} * c_{\mathrm{t\text{-}1}} + i_{\mathrm{t}} * tanh(W_{\mathrm{c}}x_{\mathrm{t}} + b_{\mathrm{c}}) \\
h_{\mathrm{t}} &= o_{\mathrm{t}} * tanh(c_{\mathrm{t}})
\end{aligned}
\tag{5.4}
$$

where σ is the sigmoid activation, and i, f, o, c are respectively the input gate, forget gate, output gate, and cell activation vectors which vector size is same as the hidden vector h. All weights are diagonal matrices.

The attention mechanism is widely combined with LSTMs in the sequential analysis that consists of two vital components, a scoring module and a weighted distribution equation, as presented in Fig. 5.3. A scoring module can be formed by various deep learning networks that take into account a hidden vector h, then output a score value. Specifically, given a sequence of hidden vector $h = \big(h_1, ..., h_{\mathrm{T}}\big)$, the scoring module generates the score value for each vector h_{t} by the following equation:

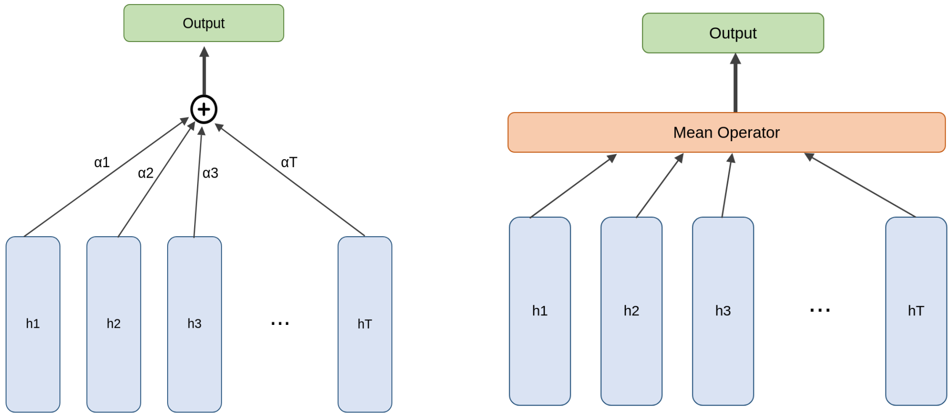
$$score(h_{\mathrm{t}}) = W_{\mathrm{s}}h_{\mathrm{t}} \tag{5.5}$$

After that, the weighted distribution Eq. 5.6 computes weights for inputs from its score value as follows:

$$\alpha_{\mathrm{t}} = softmax(score(h_{\mathrm{t}})) \tag{5.6}$$

Finally, the content vector c is computed as the weighted average over all hidden vectors as follows:

$$c = \sum_{n=1}^{T} \alpha_{\mathrm{t}} h_{\mathrm{t}} \tag{5.7}$$



(a) Using softmax for the score module

(b) Using mean for the score module

Figure 5.3: An overview of the attention mechanism.

Hidden layer feedforward network. Lastly, the model concatenates three outputs from the temporal module into a unique vector h_E and pushes it to a feedforward layer with Tanh activation. Then, the model outputs the final 128-dimensional vector representation of the encoder.

5.3.2 Decoder

The decoder consists of CNN-LSTM networks and an up-sampling unit. CNN-LSTM networks, which consist of a CNN and an LSTM component, are responsible for capturing spatio-temporal features of prediction outputs. The CNN architecture for the decoder is identical to **SC**'s model. Besides that, the up-sampling unit aims at generating heat-map images of future AQ levels. Similar to [27], the output of prediction time step t becomes the input of the next time step $t + 1$, except the first time step, whose input is the last input image of the encoder.

Up-sampling Unit. The up-sampling unit is responsible for generating heat-map im-

Table 5.3: Configurations of an up-sampling unit for AQNet

Layers	Operator	Filter Size	Output Size
Input Concatenation	Concat	-	256
Hidden	Tanh	-	256
Transform	Reshape	-	2×2×64
Transposed CNNs	ReLU	5×5	4×4×128
Transposed CNNs	ReLU	5×5	11×11×64
Transposed CNNs	ReLU	5×5	25×25×1

ages of future AQ levels. It is based on a Transposed CNN architecture (or Deconvolutional Neural Networks as referred to in some papers). Table 5.3 shows the architecture of Transposed CNNs used in the decoder. At each prediction time step t, the model concatenates the CNN-LSTM output with the hidden output h_E into a 256-dimensional vector. This vector is fed to the up-sampling unit. Since the difference of particulate matters values among adjacent districts is large, the model needs to generate various predictions among adjacent areas for valuable predictions effectively. Therefore, we used Transposed-CNN layers described in DCGAN [28] for providing sharp output images.

5.3.3 Training Algorithm

Henceforth, we denote the length of the encoder as l_E and the length of the decoder as l_D, to mean input/output time lengths respectively in our model. Algorithm 1 outlines our model's training process. First, we pre-processed collected data to make training and testing vectors. Secondly, we transformed AQI and meteorological observational data of the investigated area into heat-map images to reduce execution time. Next, we constructed training samples from the pre-processed data. At each iteration, the model takes l_E heat-map images, and feed them to the **SC** component. The outputs of this component are l_E spatial representation vectors of heat-map images. **WC** and **NC** also take l_D and l_E input vectors from their corresponding data sources. After that, all three components feed their input vectors to the temporal module. The subsequent steps of the encoder follow the flow of our model, as mentioned above. Finally, the decoder begins the processes of generating l_D heat-map images of future air quality.

Algorithm 1 AQNet Training

Input: $\mathcal{SC}, \mathcal{WC}, \mathcal{NC}$

Output: Trained AQNet Model f_θ.

```
$\mathcal{P} = \{\}$
for all available time t do
    $x_t$ = Spatial_Transform($\mathcal{SC}_t$)
    Add $x_t$ to $\mathcal{P}$
end for
Initialize all parameters $\theta$ in AQNet $f_\theta$
repeat
    for all available instance $x_t \in \mathcal{P}$ do
        $X_t$ = Spatial_Represent($\{x_i\}_{i=t-l_E}^{t}$)
        $h_I$ = Temporal_Module($X_t$)
        $h_M$ = Temporal_Module($\{\mathcal{WC}_i\}_{i=t}^{t+l_D}$)
        $h_N$ = Temporal_Module($\{\mathcal{NC}_i\}_{i=t-l_E}^{t}$)
        $h_E$ = Concatenate([$h_i$, $h_e$, $h_m$])
        $Y$ = Decode($X_t$, $h$)
        find $\theta$ by minimizing the loss function with $Y$
    end for
until Stopping criteria are met
```

Chapter 6

EXPERIMENTS

6.1 Baselines and State-of-the-art methods

Here, we briefly explain seven alternatives for comparison and divide them into two categories.

Baselines. We compared our model results with common time-series methods.

- **SVR** Support Vector Regression uses the same principles as the Support Vector Machine (SVM) with only a few differences. SVR model uses the radial basis function (RBF) kernel. The epsilon value is set to 0.1.

- **DNN** Fully-connected deep neural network consists of three layers with [128, 64, 32] hidden ReLU units.

- **CNN** Convolutional neural network consists of three 1-D convolution layers with kernel size 5, [256,128,64] filters, and pooling layers.

- **RNN** Recurrent neural network consists of [64,64] encode-decoder states with LSTM units.

State-of-the-art methods. We implemented state-of-the-art methods of time-series prediction and spatio-temporal prediction and compared their results on Seoul air pollution dataset with our proposed method.

- **DA-RNN** [29]: Dual-stage attention-based recurrent model for time-series prediction.
- **GCRN** [30]: Graph convolutional recurrent networks for predicting time-varying graph-based data using ChebNet [31]. K=2 localized filter is used for experiments.
- **DCRNN** [9]: Diffusion convolutional recurrent networks using bidirectional graph random walk. K=2 diffusion step is used for experiments.

6.2 Experimental Settings

6.2.1 Implementation details

In this part, we describe hyper-parameters and configurations of experiments. First, we set the sequence length of **SC** and **NC** as 24 and 48, while the length of **WC** is the same as the number of the decoder's time steps. In training and testing for hourly predictions, our model generates 24 hours in one execution time. To reduce execution time for long-term predictions and let our model focus on a specific prediction day, it only generates output images after an offset O. For example, we set O as 48 to stop our model from generating outputs up to 48th, and it only predicts from the 49th time step. The output of each day is the aggregated value of the corresponding hours.

Practically, we set the learning rate as 2x10^{-3} and used ADAM optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. To avoid overfitting, we set the dropout rate as 0.5 for multiple layers and applied the early stopping technique. Our model uses L_2 loss function, as shown in Eq. 6.1.

$$\mathcal{L} = \frac{1}{W}\frac{1}{H}\sum_{i=1}^{W}\sum_{j=1}^{H}(Y_{\mathrm{i,j}} - \widehat{Y}_{\mathrm{i,j}})^2 \tag{6.1}$$

where W and H are the width and height of a heat-map image, Y is the ground-truth image, while $\widehat{Y}$ is the predicted one generated from AQNet.

6.2.2 Evaluation Metric

Mean absolute error (MAE) used to evaluate model performance is defined as follows:

$$MAE = \frac{\sum_i \mid y_\mathrm{i} - \widehat{y}_\mathrm{i} \mid}{n} \tag{6.2}$$

where y_i and $\widehat{y}_\mathrm{i}$ are the real value and prediction value of time step i, and n is the total number of predictions.

6.3 Experimental Results

In this part, we present the experimental results for two datasets, as mentioned in the previous section. In experimenting with Seoul dataset, all prediction results were concentration values. With China 1-year dataset, we compared our model performance with FFA's method on Chinese AQI standards. We used the following terms to indicate the difference in time:

1. Short term: $\leq$ 8 hours

2. Middle term: 9-24 hours

3. Long term: $\geq$ 1 day

Before moving to experimental results, we demonstrate the statistical information of $PM_{2.5}$ and PM_{10} in Seoul (2014-2018) and Beijing (May 2014 - April 2015). In Seoul, we calculated the mean (μ) and standard derivation (σ) values in particle concentrations ($\mu g/m^3$). In contrast, we calculated these values in both particle concentrations and Chinese AQI standards for Beijing. As shown in Table 6.1, the mean and standard derivation of $PM_{2.5}$ in Beijing are several times greater than in Seoul. We did not compute the statistical values for PM_{10} in Beijing due to a large amount of missing data.

Table 6.1: Statistic of $PM_{2.5}$ and PM_{10} in Seoul (2014 - 2018) and Beijing (May 2014 - April 2015)

City	$PM_{2.5}$		PM_{10}		Unit
	μ	σ	μ	σ	
Seoul	24.75	16.54	45.99	34.07	$\mu g/m^3$
Beijing	83.12	80.12	-	-	$\mu g/m^3$
	106.67	91.79	-	-	China AQI

6.3.1 Performance on Spatial Module Selection

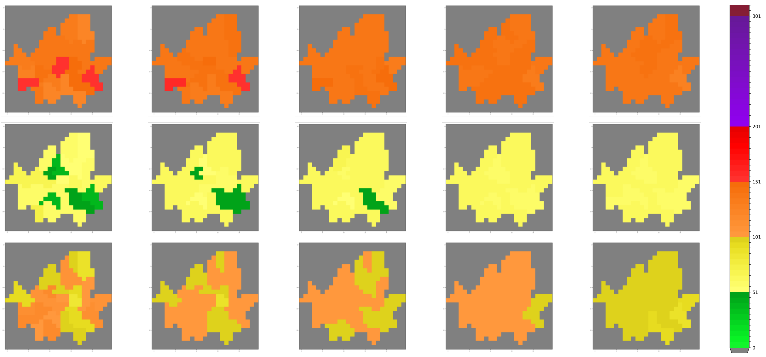
In this section, we evaluate the efficiency of using different CNN architectures in **SC** component and the decoder up-sampling units. We conducted all experiments with full data sources ($\mathcal{I}$, $\mathcal{N}$, and $\mathcal{M}$). For the **SC** component, we compared multi-scale CNNs with simple CNNs. The simple CNN architecture is a non-multi-scale version of Table 5.2. For up-sampling units in the decoder, we chose either transposed CNNs or one feedforward layer. We then set up four settings from these CNN architectures, which can be defined as follows:

1. **MS-CNNs + Trans-CNNs**: the encoder uses multi-scale CNNs, while the decoder uses Transposed CNNs.

2. **CNNs + Trans-CNNs**: the encoder uses only CNNs, a non-multi-scale version of the one above, and the decoder uses Transposed CNNs.

3. **MS-CNNs**: the decoder is the same as CNNs, and the encoder uses only multi-scale CNNs.

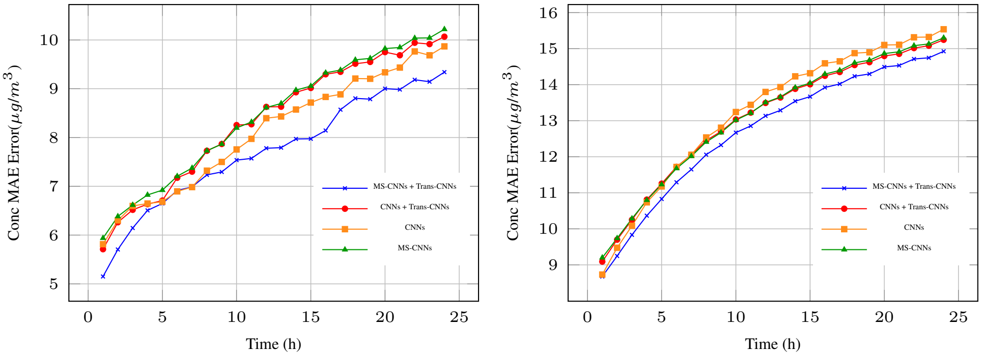
4. **CNNs**: the encoder uses only CNNs, while the decoder generates outputs, whose size is equal to 25 × 25, using only one feedforward layer with ReLU activation.



(a) Ground Truth (b) MS-CNNs + Trans-CNNs (c) CNNs + Trans-CNNs (d) CNNs (e) MS-CNNs

Figure 6.1: Examples of spatial outputs of different CNNs types of IC encoder and the generation decoder. The color bar is based on US AQI scale.

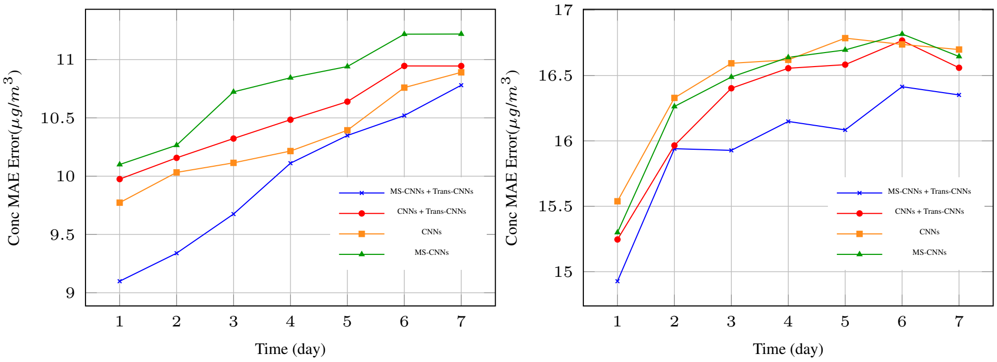
Fig. 6.1 depicts the generated outputs of each setting of CNN architectures used in our model. It shows that using transposed-CNN for up-sampling units of the decoder generates sharper images than using a simple feedforward layer. Intuitively, the generated images of MS-CNNs + Trans-CNNs are closer to the ground truth labels.



(a) $PM_{2.5}$ (b) PM_{10}

Figure 6.2: Comparison of spatial modules selection on **hourly** predictions.

As shown in Fig. 6.2, the accuracy of MS-CNNs + Trans-CNNs is superior to other settings. Fig. 6.3 elaborates the effectiveness of this module selection in long-term predictions. MS-CNNs, including various kernel sizes, can efficiently extract both local and global features in a heat-map image. Also, transposed CNNs generate high-resolution output images leading to significant improvement in prediction. Interestingly, using only CNNs in the encoder (4rd setting) also generates noticeable results compared with others. Even though the results are noticeable in terms of MAE scores, the output images are vague due to the spatially average properties of convolutional neural networks.



(a) $PM_{2.5}$

(b) PM_{10}

Figure 6.3: Comparison of spatial modules selection on **long-term** predictions.

Since we aimed at predicting future air quality spatio-temporally, the accuracy was not the only concern for our approach. We also focused on providing reliable predictions for all districts in an investigated city. Therefore, we chose multi-scale networks for spatial feature extraction and transposed CNNs for up-sampling units of the decoder to achieve the best performance.

6.3.2 Comparison to Baselines and State-of-the-art Methods

We compared our model with baseline and state-of-the-art models in hourly prediction for up to 24 hours, and long-term predictions for up to seven days ahead.

Table 6.2: Comparison of Hourly Predictions of **PM$_{2.5}$**

Methods	1-3h	4-6h	7-9h	10-12h	13-18h	19-24h
SVR	6.73	7.85	8.63	9.32	9.94	10.65
DNN	4.17	6.53	7.95	8.86	9.78	10.57
CNN	4.02	6.47	7.87	8.85	9.81	10.67
RNN	4.03	6.52	7.93	8.85	9.81	10.63
DA-RNN	5.81	7.17	8.11	8.85	9.61	10.28
GCRN	5.50	6.86	7.92	8.63	9.39	10.08
DCRNN	5.56	6.87	7.88	8.61	9.40	10.04
AQNet	5.67	6.69	**7.17**	**7.63**	**8.21**	**9.07**

Table 6.3: Comparison of Hourly Predictions of **PM$_{10}$**

Methods	1-3h	4-6h	7-9h	10-12h	13-18h	19-24h
SVR	15.77	16.50	17.12	17.59	17.98	18.47
DNN	6.61	10.11	12.23	13.47	14.71	15.95
CNN	5.85	9.90	12.17	13.64	14.91	16.16
RNN	5.96	10.00	12.22	13.6	14.87	15.96
DA-RNN	9.47	11.78	13.34	14.39	15.3	16.18
GCRN	8.52	11.35	13.22	14.41	15.41	16.21
DCRNN	8.17	11.12	13.01	14.18	15.23	16.07
AQNet	9.25	10.83	**12.01**	**12.89**	**13.78**	**14.62**

In hourly prediction, our model performs better than baselines and state-of-the-art methods in the middle term as presented in Tables 6.2 and 6.3. On the contrary,

Table 6.4: Comparison of Long-Term Predictions of $\mathbf{PM_{2.5}}$

Methods	1d	2d	3d	4d	5d	6d	7d
SVR	10.32	12.35	12.51	12.41	12.75	12.76	12.38
DNN	10.69	11.78	11.94	11.97	11.91	12.06	12.02
CNN	10.70	11.82	12.03	11.99	11.99	12.14	12.03
RNN	10.74	11.66	11.96	12.04	11.96	12.16	12.07
DA-RNN	10.45	11.40	11.60	11.80	11.56	11.44	11.37
GCRN	10.40	11.36	11.53	11.43	11.27	11.23	11.26
DCRNN	10.18	11.48	11.57	11.44	11.28	11.24	11.31
AQNet	**9.10**	**9.34**	**9.67**	**10.11**	**10.35**	**10.52**	**10.78**

Table 6.5: Comparison of Long-Term Predictions of $\mathbf{PM_{10}}$

Methods	1d	2d	3d	4d	5d	6d	7d
SVR	18.41	19.63	19.83	19.91	19.99	19.90	20.07
DNN	16.13	17.71	17.82	17.59	17.91	18.15	18.24
CNN	15.90	17.46	17.58	17.28	17.69	18.08	17.80
RNN	16.35	17.54	17.55	17.66	17.71	18.23	18.03
DA-RNN	17.22	18.60	19.01	19.04	19.31	19.04	18.46
GCRN	16.27	17.60	17.97	18.13	18.33	18.43	18.43
DCRNN	16.09	17.60	17.97	18.33	18.52	18.63	18.65
AQNet	**14.93**	**15.94**	**15.93**	**16.15**	**16.08**	**16.41**	**16.35**

baseline models, especially time-series methods, outperformed our proposed method in the short term.

In long-term predictions, AQNet is superior to baselines and state-of-the-art methods, as shown in Tables 6.4 and 6.5. To compare our method with baselines and state-of-the-art methods, we averaged MAE scores of seven days for all methods. It showed that the proposed method 15.2% and 8.2% improvement in MAE scores for $PM_{2.5}$ and PM_{10}, respectively.

The variations in PM_{10} and $PM_{2.5}$ strongly correlate with the changes in time due to the complex interactions of various factors. In the middle and long term, these interactions cause significant variations in AQ levels, especially spatially. As a result, transforming monitoring data into heat-map images lets our model capture spatial relations efficiently leading to superiority when predicting air quality for a more extended period. However, due to the spatial learning process, our model is less sensitive to the short-term variations, which affects the accuracy in short-term predictions. Therefore, in the short term, time-series methods such as DNN, CNN, and RNN perform better than AQNet.

In training and testing, all baseline models use only local AQ data as input, while our model uses various data sources. We also conducted experiments on baselines with multiple data sources. However, adding more data sources to baseline models degrades their accuracy. It reveals the weakness of baselines in term of assessing the impact of critical factors on AQ prediction.

6.3.3 Evaluation on China 1-year Dataset

The previous method, proposed in Zheng et al., 2015 (FFA), is an ensemble model based on basic machine learning algorithms. It uses 2:1 ratio for data separation in the experiments. For example, it tests our model with 3, 6, 9, 12, and training with the other months. In the paper, the experimental results are only the predictions of $PM_{2.5}$ levels in Chinese AQI standards for Beijing.

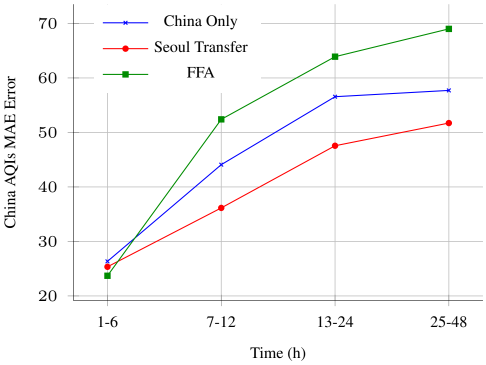
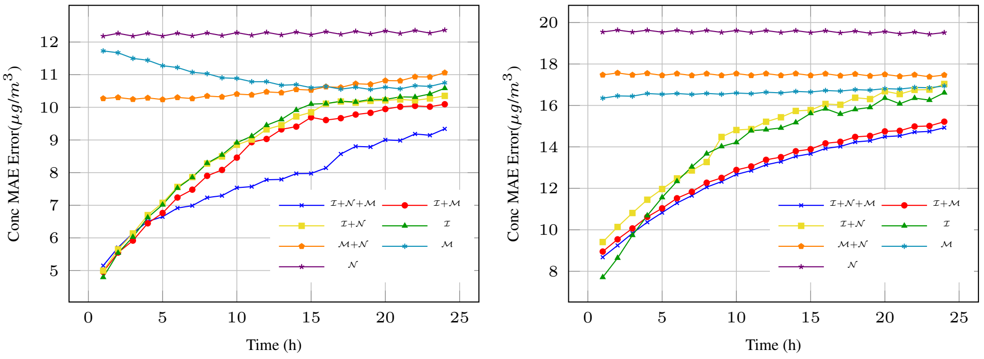


Figure 6.4: Compared with FFA on prediction results of $PM_{2.5}$ levels in Chinese AQI standards.

We conducted two experiments on this dataset. First, we evaluated our model robustness with an experiment, which trains our model from scratch, named as China Only. In the China-only setting, we also trained our model for 8 months and tested with 4 months as similar to the previous method. Secondly, we checked our model's capability of dealing with the lack of data by applying transfer learning on Seoul dataset to China dataset, called Seoul Transfer. In this experiment, we pre-trained our model on Seoul dataset for 20 epochs then re-trained our model on China dataset for 300 epochs. Both settings outperformed the previous method, and the transfer learning setting is superior to others, as shown in Fig. 6.4. It points out that transfer learning is appropriate for dealing with the lack of data in China 1-year dataset. However, the results also reveal our model weakness for short-term predictions.

6.3.4 Assessing the Impact of Critical Factors

According to [3], the dispersion and formation of particulate matters depend on time, wind speed, precipitation, and humidity. To assess the impact of critical factors on future AQ levels, we performed experiments on all possible combinations of data sources $\mathcal{I}$, $\mathcal{M}$, and $\mathcal{N}$. These combinations can be divided into two categories (1) involving $\mathcal{I}$ and (2) canceling $\mathcal{I}$ due to the similar trend of prediction errors.



(a) PM$_{2.5}$

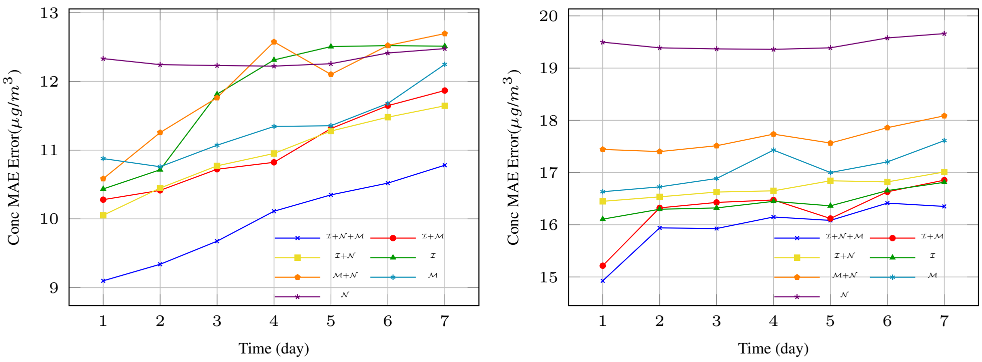
(b) PM$_{10}$

Figure 6.5: An assessment of critical factors affecting the **hourly** prediction.

Short-to-middle-term impact. As shown in Fig. 6.5, adding local AQ data ($\mathcal{I}$) to the predictive model (E.g., $\mathcal{I}+\mathcal{M}$, $\mathcal{I}+\mathcal{N}$) shows superior performance compared to the settings when removing $\mathcal{I}$ from the input data. We can infer that short-to-middle-term AQ predictions firmly rely on local AQ data.

As depicted in Fig. 6.5a, the performance of predicting AQ levels using only meteorology data ($\mathcal{M}$) is poor in the short term, while it gradually improves in the middle term. It reveals the increased influence of meteorology on PM$_{2.5}$ in the middle term. The parallel performance of experimental settings with $\mathcal{I}$, $\mathcal{I}+\mathcal{M}$, and $\mathcal{I}+\mathcal{N}$ indicate the corresponding impact of both local meteorological conditions and neighboring air pollution sources on short-to-middle-term PM$_{2.5}$ levels. Also, the best performance of the experiment with all data sources emphasized the strong correlation of PM$_{2.5}$ variations with critical factors. Besides, the weak results of the setting using only $\mathcal{N}$ data reveal that the external air pollution sources have less influence on short-to-middle-term PM$_{2.5}$ levels than meteorological conditions. Therefore, short-to-middle-term PM$_{2.5}$ levels are the results of a combination of critical factors.

Similar to PM$_{2.5}$, PM$_{10}$ levels are also connected with critical factors in the short and middle term. However, the impact of each factor on short-to-middle-term PM$_{10}$ levels is not as discernible as PM$_{2.5}$. As Fig. 6.5b shows, the errors of experimental



(a) $PM_{2.5}$

(b) PM_{10}

Figure 6.6: An assessment of critical factors affecting the **long-term** prediction.

settings $\mathcal{N}$, $\mathcal{M}$, and $\mathcal{M}+\mathcal{N}$ are constant and significant. Also, the performance of two settings $\mathcal{I}+\mathcal{N}$ and $\mathcal{I}$ are similar, which emphasizes the negligible impact of the external air pollution sources on the variations of short-to-middle-term PM_{10}. Finally, the analogy of the performance of two settings $\mathcal{I}+\mathcal{N}+\mathcal{M}$ and $\mathcal{I}+\mathcal{M}$ shows the strong correlation of meteorology and local observational AQI with PM_{10} in the short and middle term.

Long-term impact. From Fig. 6.6, we can see the specific influence of critical factors on long-term air quality.

As depicted in Fig. 6.6a, meteorology and the external air pollution sources significantly impact the variations of long-term $PM_{2.5}$ levels. First, all experimental settings without the advent of $\mathcal{I}$ have similar performance with the prediction setting using only $\mathcal{I}$ data. Next, the comparable performance of $\mathcal{M}+\mathcal{N}$, $\mathcal{I}+\mathcal{M}$, and $\mathcal{I}+\mathcal{N}$ indicate the relative effects of critical factors on long-term $PM_{2.5}$ levels. Finally, similar to short-to-middle-term predictions, the distinctive results of the combination of all data sources emphasize the strong interconnectedness of meteorology, the external air pollution sources, and the future $PM_{2.5}$ levels.

Unlike long-term $PM_{2.5}$ predictions, the impact of decisive factors on PM_{10} level predictions are not evident, especially the external air pollution sources. From Fig.

6.6b, the experiment with only $\mathcal{N}$ data gives the worst performance followed by the settings $\mathcal{M}+\mathcal{N}$ and $\mathcal{M}$. The performance of the predictive model using only local AQ data ($\mathcal{I}$) is a bit worse than the results of the experiment with $\mathcal{I}$ and $\mathcal{N}$. Similar to short-to-middle-term predictions, it shows the slight influence of the external air pollution sources on PM$_{10}$ levels. Also, the analogy of results of the two settings $\mathcal{I}+\mathcal{N}+\mathcal{M}$ and $\mathcal{I}+\mathcal{M}$ indicate the direct influence of meteorology on the variations of long-term PM$_{10}$. Finally, the distinguished performance of all data sources combination confirm the accumulative impacts of all factors on PM$_{10}$ levels.

To summarize, PM$_{2.5}$ is sensitive to both meteorological conditions and external air pollution sources. In contrast, PM$_{10}$ is more localized and is less affected directly by external air pollution sources than PM$_{2.5}$. Furthermore, the influences of these factors on air quality level are more significant in the middle and long term.

Chapter 7

CONCLUSION

In this thesis, we proposed a data-driven approach, which extracts and combines features from various data sources, to spatio-temporal air quality prediction. Additionally, we developed a deep generative model for predicting short-to-long-term air quality levels. Unlike previous methods, our model transforms local air quality observations into heat-map images to capture spatial relations of air quality levels efficiently. The experimental results showed that our model outperforms baselines and state-of-the-art methods, especially in long term predictions. The results on China 1-year dataset demonstrated our model robustness in both the training from scratch and the transfer learning setting. These results open the door for dealing with the lack of data for air quality prediction of many urban areas. Next, we assessed the impact of specific factors on future air quality. Besides that, we realized the weaknesses of an LSTM-based model, which usually makes predictions based on previous values. However, future air quality does not entirely depend on previous time-steps due to the strong interconnectedness of air quality levels and many factors. As a result, we plan to focus on air quality prediction for sudden changes along with short-to-long-term predictions. Also, we will extend the proposed model to an IoT platform.

Bibliography

[1] Y. Kim, J. Seo, J. Y. Kim, J. Y. Lee, H. Kim, and B. M. Kim, "Characterization of pm 2.5 and identification of transported secondary and biomass burning contribution in seoul, korea," *Environmental Science and Pollution Research*, vol. 25, no. 5, pp. 4330–4343, 2018.

[2] Y. P. Kim and G. Lee, "Trend of air quality in seoul: Policy and science," *Aerosol Air Qual. Res*, vol. 18, pp. 2141–2156, 2018.

[3] WHO, "Health risks of particulate matter from long-range transboundary air pollution," vol. Regional Office for Europe & Joint WHO/Convention Task Force on the Health Aspects of Air Pollution Copenhagen : WHO Regional Office for Europe, 2016.

[4] C. A. Pope III, R. T. Burnett, M. J. Thun, E. E. Calle, D. Krewski, K. Ito, and G. D. Thurston, "Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution," *Jama*, vol. 287, no. 9, pp. 1132–1141, 2002.

[5] M. Kampa and E. Castanas, "Human health effects of air pollution," *Environmental pollution*, vol. 151, no. 2, pp. 362–367, 2008.

[6] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[8] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[9] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[10] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artificial Intelligence*, vol. 259, pp. 147–166, 2018.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[13] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[14] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1419–1426.

[15] Y. Qi, Q. Li, H. Karimian, and D. Liu, “A hybrid model for spatiotemporal forecasting of pm2. 5 based on graph convolutional neural network and long short-term memory,” *Science of the Total Environment*, vol. 664, pp. 1–10, 2019.

[16] Y. Lin, N. Mago, Y. Gao, Y. Li, Y.-Y. Chiang, C. Shahabi, and J. L. Ambite, “Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning,” in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2018, pp. 359–368.

[17] J. Y. Zhu, Y. Zheng, X. Yi, and V. O. Li, “A gaussian bayesian model to identify spatio-temporal causalities for air pollution based on urban big data,” in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2016, pp. 3–8.

[18] J. Y. Zhu, C. Sun, and V. O. Li, “Granger-causality-based air quality estimation with spatio-temporal (st) heterogeneous big data,” in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2015, pp. 612–617.

[19] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, “Support vector regression machines,” in *Advances in neural information processing systems*, 1997, pp. 155–161.

[20] Y. Zheng, F. Liu, and H.-P. Hsieh, “U-air: When urban air quality inference meets big data,” in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1436–1444.

[21] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, “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, 2015, pp. 2267–2276.

[22] W. Cheng, Y. Shen, Y. Zhu, and L. Huang, “A neural attention model for urban air quality inference: Learning the weights of monitoring stations,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[23] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, “Deep distributed fusion network for air quality prediction,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 965–973.

[24] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, “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, 2015, pp. 2267–2276.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[26] ——, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[27] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[28] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

[29] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction,” *arXiv preprint arXiv:1704.02971*, 2017.

[30] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, “Structured sequence modeling with graph convolutional recurrent networks,” in *International Conference on Neural Information Processing*. Springer, 2018, pp. 362–373.

[31] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

초록

세계 경제 활동과 에너지 수요가 증가함에 따라 많은 국가들이 대기 오염에 대한 우려를 제기하고 있다. 하지만 많은 요인들의 복잡한 상호 작용으로 인해 대기 질을 예측하는 것은 어려운 문제다. 본 논문에서는 AQNet이라는 이름의 시공간적 대기 질 예측을 위한 심층 생성 모델을 제안한다. 이전 연구와 달리 이 모델은 대기 질 지수 데이터를 2D 프레임(히트 맵 이미지)으로 변환하여 대기 품질 수준의 영역 간 공간적 관계를 효과적으로 포착한다. 그런 다음 기상과 외부 대기 오염원과 같은 중요한 요소의 시간적 특징과 공간 표현을 결합한다. 예측 모델은 먼저 미래의 대기 품질 수준의 히트 맵 이미지를 생성한 다음 해당 영역의 출력 값으로 집계한다. 데이터 분석을 토대로 대기 오염 예측에 각 주요 요소들이 미치는 영향을 평가하였다. 제안된 방법을 평가하기 위해 실제 대기 오염 데이터 세트인 서울의 데이터 세트와 중국의 1년 데이터 세트를 실험했다. 본 논문에서 제안한 방법은 서울 데이터세트에서 수행된 PM2.5와 PM10의 장기 예측에 대해 이전의 SOTA 방법과 비교하여 MAE 점수가 각각 15.2%, 8.2% 향상되었다. 또한 중국 데이터 세트에 대한 이전 연구와 비교하여 PM2.5 예측의 MAE 점수를 20% 향상시켰다.

주요어: 서울대학교 논문양식, TeX

학번: 2017-20722

ACKNOWLEDGEMENT

I would like to thank my family, my wife, my advisor, friends, and colleagues for their outstanding support. To my life-coach, my parents: because I owe it all to you. To my wife, I owe you many thanks for your understanding and sympathy during the last three years. I am grateful to my advisor, Professor Sang Kyun Cha, for his valuable advises and for the funding for my research. I am thankful to Professor Kyomin Jung and Professor Insoon Yang for their comments and assistance during this research. I am also grateful to the following university friends: for their unfailing support and help. Finally, last but by no means least, to everyone in PIDL lab, it was excellent sharing laboratory with all of you during the last three years.