

Xie Yu

Deep Learning Architectures for PM2.5 and Visibility Predictions

Technische Universiteit Delft



Deep Learning Architectures for PM2.5 and Visibility Predictions

By

Xie Yu

in partial fulfilment of the requirements for the degree of

Master of Science

in Applied Mathematics

at the Delft University of Technology,

to be defended publicly on Tuesday August 28, 2018 at 13:30.

Supervisor: Prof. dr. ir. H.X.Lin

Msc. J. Jin

Thesis committee: Prof. dr. ir. H.X.Lin

Dr. Ir. M.B. van Gijzen

Dr. B.J. Meulenbroek

This thesis is confidential and cannot be made public until August 27, 2018.

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Facing the severe air pollution phenomenon in urban areas and the subsequent low visibility event in airports, it is urgent to conduct air quality and visibility predictions to better reflect their changing trends. However, the variations of PM2.5 and visibility involve complicated physical and chemical processes, which make their accurate predictions challenging.

In this thesis, methodologies to predict PM2.5, PM10, and visibility using Long Short-Term Memory Neural Networks (LSTM NN) were investigated. The first step of the proposed methodology was dataset analysis and preprocessing, which is an important step in almost all machine learning problems. Because missing data and confusing or incorrect data are common in large datasets, noise and errors were corrected and missing rates were calculated at first. Afterward, datasets were visualized to evaluate the missing phenomenon of different features. Due to the explored strong spatiotemporal correlations, for air quality features with high missing rates, linear interpolations were implemented when the missing granularity is small and k-Nearest Neighbor (kNN) imputations were used when the missing interval is large.

Furthermore, the PM2.5 or PM10 prediction is usually considered as a regression task and aimed at minimizing the mean squared error (MSE) between the predicted values and measured ones. However, due to the high variability and explored 'class-imbalance' phenomenon of visibility data, that is, most of the data we have are related to 'normal' situations and extreme conditions are rare events, its predictions can be better dealt with as a classification problem. Because the most interesting cases to be predicted are those rare extreme events, the target was adapted to minimize the weighted cross-entropy.

The second step of the proposed methodology was to configure the frameworks. For PM2.5 predictions, feature engineering was employed to select appropriate features and some model hyperparameters were set through grid searches and coordinate descent. A coarse-to-fine sampling scheme was used to determine the weights in the loss function of visibility predictions.

The third step of our research was performance evaluation. For PM2.5 predictions, the proposed spatiotemporal LSTM framework can overcome the systematic underestimation that Lotos-Euros (a chemical transport models (CTMs) based system) generally produces by analyzing their scatter plots and confusion matrices. Additionally, it performs better than an LSTM-based prediction framework (Fan J et al. (2017) [9]) that also considers spatial correlations among stations and performs a similar task in a similar region when comparing their rooted mean square errors (RMSE) and mean absolute errors (MAE). Differences between the hyperparameters of these two frameworks were analyzed.

As for PM10 predictions, the training efficiency can be improved significantly by transferring knowledge from PM2.5 predictions to PM10 predictions through model fine-tuning. Compared with Lotos-Euros, the LSTM framework also has competitive performance in PM10 predictions. As the first attempt at applying LSTM NN to predict visibility, forecasts are acceptable in practice. The total accuracy rate reaches 90.61%. The recall rate of the normal situation (L1) is 93% while its precision rate is 96%, indicating its superior prediction performance in the normal situations. Besides, for each visibility level, the number of correct predictions is larger than that of negative predictions.

Keywords: PM2.5 predictions, PM10 predictions, Visibility predictions, Deep learning, LSTM, Transfer learning

Preface

This thesis is the final part of my master's project in Applied Mathematics at TU Delft. The thesis project was done in the Mathematical Physics Group at the Delft Institute of Applied Mathematics.

First of all, I would like to sincerely thank my daily and responsible supervisor Prof. dr. ir. H.X.Lin. He provided me with a lot of guidance and help during this thesis project. Without his professional suggestions, I could not complete this research so successfully. Besides, I would also like to thank MSc. J.Jin, who supported me with the data of Lotos-Euros and encouraged me for framework generalization. It was my great pleasure to work with you.

What's more, I would like to give thanks to the committee members, Dr. Ir. M.B. van Gijzen and Dr. B.J. Meulenbroek for participating in the evaluation of this research project.

I would also like to thank my dear friends, Siyu Guan and Yu Rong. Their support and accompany makes my life in the Netherlands much more fun. With their encouragements and help, I feel inspired all the time.

Finally, I would like to express many thanks to my dear boyfriend, Ma Xu, and my parents, who support and encourage me all the time!

Xie Yu

Delft, August 2018

Contents

1. INTRODUCTION	1
1.1 CHALLENGES OF PM2.5 PREDICTIONS.....	1
1.2 CHALLENGES OF VISIBILITY PREDICTIONS.....	3
1.3 RESEARCH OBJECTIVES.....	3
1.4 RESEARCH SCOPE.....	4
2. LITERATURE STUDY	7
2.1 INTRODUCTION TO SPATIOTEMPORAL DATA MINING.....	7
2.1.1 <i>Spatiotemporal Data Properties</i>	8
2.1.2 <i>Spatiotemporal Predictive Learning</i>	8
2.2 CHEMICAL TRANSPORT MODELS FOR PM2.5 PREDICTIONS	9
2.2.1 <i>Lotos-Euros</i>	10
2.2.2 <i>CMAQ</i>	11
2.3 TRADITIONAL STATISTICAL METHODS FOR PM2.5 PREDICTIONS	12
2.4 DEEP LEARNING METHODS FOR PM2.5 PREDICTIONS.....	12
2.4.1 <i>Autoencoders</i>	13
2.4.2 <i>Deep Belief Networks</i>	13
2.4.3 <i>Recurrent Neural Networks</i>	14
2.4.4 <i>Long Short-Term Memory Neural Networks</i>	14
2.4.5 <i>Hybrid Models</i>	16
2.5 RELATED WORK ON VISIBILITY PREDICTIONS.....	16
2.5.1 <i>Numerical Weather Prediction</i>	17
2.5.2 <i>Statistical Methods</i>	17
2.6 SUMMARY	18
3. THEORETICAL BACKGROUND FOR DEEP LEARNING ALGORITHMS	21
3.1 RECURRENT NEURAL NETWORKS	21
3.2 LSTM NN FORMULATION.....	23
3.3 DROPOUT IN RECURRENT NEURAL NETWORKS	24
3.4 GRADIENT DESCENT OPTIMIZATION ALGORITHMS	27

3.4.1 <i>Mini-batch Gradient Descent</i>	27
3.4.2 <i>Adam</i>	28
3.5 PERFORMANCE MEASURES	29
3.5.1 <i>Regression Task</i>	30
3.5.2 <i>Classification Task</i>	31
3.6 SUMMARY	33
4. METHODOLOGY FOR PM2.5 AND VISIBILITY PREDICTIONS USING DEEP LEARNING	35
4.1 PROBLEM DESCRIPTION OF PM2.5 PREDICTIONS	35
4.2 PROBLEM DESCRIPTION OF VISIBILITY PREDICTIONS.....	36
4.3 DATASET FOR PM2.5 PREDICTIONS	36
4.3.1 <i>Meteorological Data Visualization</i>	37
4.3.2 <i>Air Quality Data Visualization</i>	39
4.3.3 <i>Spatiotemporal Correlation Analysis</i>	40
4.4 DATASET FOR VISIBILITY PREDICTIONS.....	42
4.5 DATA PREPROCESSING.....	43
4.6 SUMMARY	44
5. PERFORMANCE EVALUATION	47
5.1 PM2.5 PREDICTIONS AT STATION GUANYUAN, BEIJING	47
5.1.1 <i>Mini-Batch Gradient Descent Hyperparameters</i>	47
5.1.2 <i>Feature Selections</i>	49
5.1.3 <i>Model Hyperparameters</i>	50
5.1.4 <i>LSTM against Lotos-Euros</i>	53
5.1.5 <i>Comparison with Other Spatiotemporal Prediction Framework</i>	60
5.2 TRANSFER LEARNING FOR PM10 PREDICTIONS	62
5.3 VISIBILITY PREDICTIONS AT BEIJING CAPITAL INTERNATIONAL AIRPORT	63
5.3.1 <i>Dataset Partition</i>	64
5.3.2 <i>Framework Parameters</i>	65
5.3.3 <i>Prediction Results</i>	67
5.4 SUMMARY	67

6. CONCLUSIONS AND FUTURE RESEARCH.....	69
6.1 RESEARCH WORK RECAP	69
6.1.1 <i>Dataset Analysis and Preprocessing</i>	69
6.1.2 <i>Framework Configuration</i>	70
6.1.3 <i>Performance Evaluation</i>	70
6.2 CONCLUSIONS	71
6.2.1 <i>Research Objective 1</i>	71
6.2.2 <i>Research Objective 2</i>	72
6.2.3 <i>Research Objective 3</i>	73
6.3 RECOMMENDATIONS FOR FUTURE RESEARCH.....	74
6.3.1 <i>Prototype for Station Selections in China</i>	74
6.3.2 <i>Framework Generalization</i>	76
6.3.3 <i>Framework Update</i>	76
REFERENCES.....	77

1. Introduction

With the rapid development of economy and the process of urbanization, many serious environmental pollution problems such as air pollution, water pollution, and noise pollution arose. Among these problems, air pollution has received increasing attention worldwide in the last decades because many developing countries have suffered from serious air pollution. Take China for example, many extreme air pollution events happened, especially in the Beijing, Tianjin, and Hebei districts (Jing-Jin-Ji area) [1][2]. According to the Reports on the Chinese Air Quality State (Dec 2017) (<http://www.cnemc.cn/>), among 338 monitored cities, the percentage of days that below the national healthy air quality standard reached 34.0% on average.

As for the origins of air pollution, reactive gases and fine particles are two main sources. Higher level concentrations of these aerosols can lower visibility and increase the frequency of hazy days[3]. In the meanwhile, the occurrence of low visibility weather can cause a wide range of airport delays and cancellations[4]. This not only brings huge losses for the airlines, but also affects the transit trip. In addition, visibility is closely related to flight safety. Low visibility is one of the most common causes of flight accidents[4].

Facing the severe air pollution phenomenon in urban areas and the subsequent low visibility event in airports, it is urgent to conduct air quality and visibility predictions to better reflect their changing trends. Timely visibility predictions can help the airport operators to take measures to reduce the economic loss and passengers inconvenience caused by the air traffic disruption of low visibility. With the help of air quality predictions, governments and environmental agencies are able to enact policies and provide services to protect their citizens[2].

1.1 Challenges of PM2.5 Predictions

Air pollution indicates the introduction of particulates, biological molecules or other harmful materials into the Earth's atmosphere. It can cause disease to humans, destruction to other living organisms and also damage to the natural environment[5].

An air pollutant is defined as a substance in the air that can affect humans and the ecosystem negatively. As illustrated above, air pollution is mainly due to reactive gases and fine particles. According to the World Meteorological Organization (WMO) (www.wmo.int), reactive gases as a group are very diverse and include surface ozone (O₃), carbon monoxide (CO), volatile organic compounds (VOCs), oxidised nitrogen compounds (NO_x, NO_y), and sulfur dioxide (SO₂). Parts of reactive gases (O₃, CO, NO₂, and SO₂) adding fine particulate matters (PM2.5) and lead form the six “criteria pollutants” for air pollution. Among these substances, O₃, PM2.5, and NO₂ are the most widespread health threats[5].

PM2.5 represents fine particulate matters that consist of solid or liquid particles and are smaller than 2.5 micrometers in diameter. It can traverse the nasal passages during inhalation and reach the throat and even the lungs[6]. People who are exposed to ambient PM2.5 for a long time are likely to suffer from respiratory and cardiovascular diseases and some other illnesses[6].

The widespread sources of PM2.5 include industrial processes, energy production from power stations, vehicular traffic, residential heating, transport, natural disasters, coupled with complicated physical and chemical processes[1]. Hence, the determinants of PM2.5 concentrations include:

- Weather patterns
- Wind
- Instability
- Turbulence
- Precipitation
- Topography
- Temperature of gases

Due to these complex underlying interactions, the PM2.5 forecast remains a hard task.

1.2 Challenges of Visibility Predictions

According to the International Civil Aviation Organization (ICAO) (www.icao.int), visibility is defined as the greatest distance at which a black object of suitable dimensions, situated near the ground, can be seen and recognized when observed against a bright background. On the basis of previous studies, the size, chemical composition, and mass concentration of airborne particles substantially lead to the variation of visibility[7].

In the meanwhile, according to WMO (www.wmo.int), fog is defined as the reduction in horizontal visibility to less than 1000 m. When the observed horizontal visibility is at least 1000 m, but not more than 5000 m, the phenomenon is called mist. However, the meteorological causes of mist or fog are hard to be determined[8]. Many meteorological features such as relative humidity, pressure, wind, and temperature can directly or indirectly contribute to the degradation of visibility[7].

To summarize, visibility variations are mainly influenced by airborne particles and weather patterns. However, as illustrated in section 1.2, fine particulates like PM_{2.5} are hard to be forecasted because of the complex underlying interactions. In addition, since weather patterns are chaotic, it is still a challenging task to predict weather accurately and precisely. *Because of these two difficulties, up to now, there has been no mature physical or mathematical model for the visibility. Its prediction remains a challenging task.*

1.3 Research Objectives

The primary objective of this research is determined as follows:

To develop deep learning architectures that can predict PM_{2.5} and visibility based on historical meteorological and air pollutants information.

This primary objective can be further divided into several sub-objectives as follows:

1. To develop frameworks that can predict PM_{2.5} concentrations and visibility several hours in advance, state of the art methods are considered to accomplish this objective.

2. To explore the configuration of the chosen state-of-the-art method that can achieve the best performance.
3. To compare the methodology performance with other state-of-the-art methods in terms of root mean square errors (RMSE), mean absolute errors (MAE), confusion matrices and scatter plot diagrams.

This thesis is organized in the following way. Chapter 1 introduces the problem background, research challenges, objectives, and scope. Chapter 2 discusses the literature in PM2.5 and visibility predictions. Chapter 3 introduces Recurrent Neural Networks (RNN), the network regularization technique dropout and the network optimization algorithm Adam. Chapter 4 describes the PM2.5 and visibility prediction problem in details and presents the dataset visualization results, statistical summaries and data preprocessing method. Chapter 5 evaluates the performance of the proposed frameworks in PM2.5, PM10 and visibility predictions separately through the case study. Chapter 6 presents conclusions, discussions and further research. Figure 1.1 summarizes the outline of this report.

1.4 Research Scope

The research focuses on developing deep learning architectures for PM2.5 and visibility predictions as discussed in section 1.3. Since different approaches can be used to achieve these objectives, the scope of this research should be limited.

1. Limiting input features: Many features are available and can be considered as input for the frameworks. In this research, only meteorological and air quality features that are crucial in PM2.5 predictions are considered. In addition, some characteristics specific to the analyzed area are taken into account.

2. State of the art approaches: While many neural networks can be applied to solve the problem, in this research, the focus is on long short-term memory neural networks (LSTM NN). The reason is that this research requires performing prediction tasks from spatiotemporal data, which makes LSTM NN the most suitable approach[6]. Moreover, many LSTM-based algorithms have been performed in similar tasks of atmospheric science with very good performance[2][6][9]. Reasons

for using LSTM NN will be illustrated much more specifically in section 2.6.

3. Study area: Due to the severe air pollution phenomenon and frequent low visibility event in Jing-Jin-Ji districts, the research area is limited to Beijing.

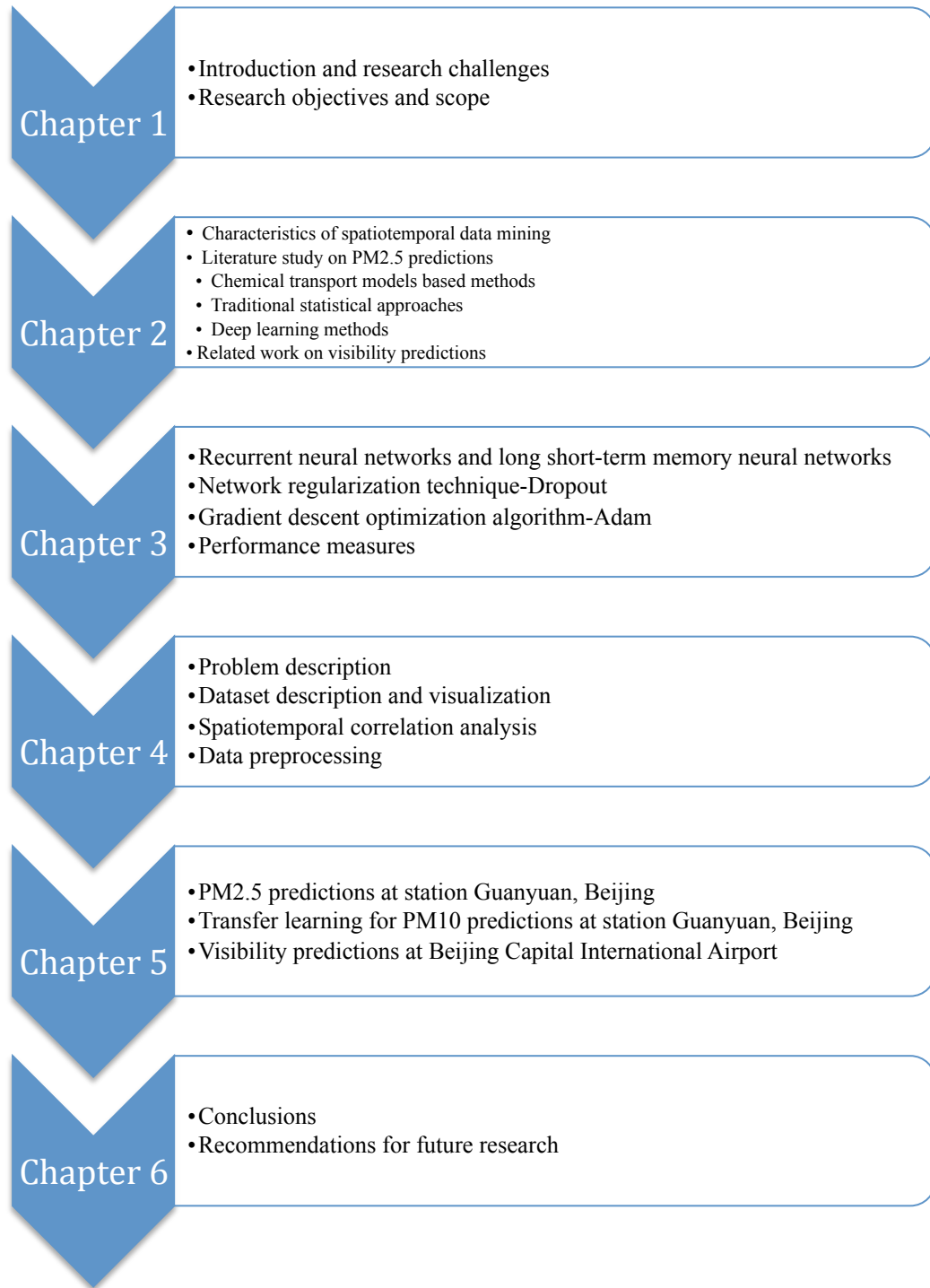


Figure 1.1 The outline of the thesis report.

2. Literature Study

In recent years, many efforts have been made to enrich approaches for forecasting air pollutant concentrations and visibility. For air pollutant predictions, methods are generally classified into two categories: Chemical Transport Models (CTMs) based and data based methods. CTMs include mechanism models that track how air pollutants generate, disperse and transmit while numerical simulations are used to produce predictive results. As for data based methods, they avoid sophisticated theoretical models and predict air quality based on data[10]. In this review, data based methods will be classified into two parts: traditional statistical methods and deep learning methods. When it comes to visibility predictions, numerical weather prediction (NWP) approaches and statistical methods are widely used[8].

2.1 Introduction to Spatiotemporal Data Mining

Because PM_{2.5} data show strong correlations in both time and space, which will be illustrated in section 4.3.3, in this research, instead of time series, we deal with spatiotemporal data. Spatiotemporal data mining aims at recognizing interesting and useful patterns from large spatiotemporal datasets[11]. Figure 2.1 shows the general procedure of spatiotemporal data mining. For a given dataset, the first step is data preprocessing to correct noise and errors, impute missing data and implement spatiotemporal analysis so as to understand the underlying interactions. Afterward, the preprocessed data are fed into an appropriate algorithm to give target patterns. Since our research focuses on predictions, the output patterns here are predictive variables.

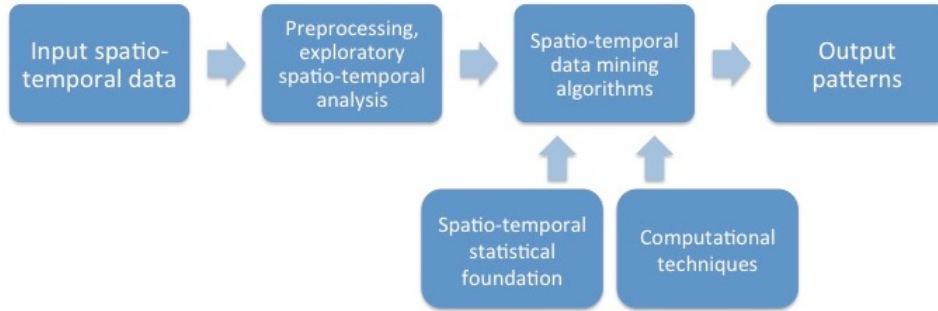


Figure 2.1 The procedure of spatiotemporal data mining (redraw based on [11]).

2.1.1 Spatiotemporal Data Properties

Since air pollutants for all areas are highly correlated, it is important to fully take spatiotemporal correlations into account. Spatiotemporal (ST) data refers to the spatial and temporal information of every measurement. Two inherent properties of ST data introduce challenges as well as opportunities for classical data mining algorithms[12].

1. Auto-correlation. In terms of ST data, the observations obtained at nearby sites and timestamps are correlated rather than independent with each other. This auto-correlation leads to a coherence of spatial observations and smoothness in temporal observations.

2. Heterogeneity. For ST data, heterogeneity in both space and time can be shown in various ways and levels. Due to this heterogeneity, different models are learned in different spatiotemporal regions.

2.1.2 Spatiotemporal Predictive Learning

Predictive learning as a kind of spatiotemporal data mining, its basic objective is to learn a mapping from the input features to the output variables through a representative training set[11].

While traditional methods are able to identify temporal characteristics of input features, novel techniques considering spatial information in ST rasters are desired. In our research, *information about spatial neighborhoods is leveraged to ensure spatial consistency among values at nearby sites*. Variants of recurrent neural networks including spatial information for spatiotemporal predictive learning have been investigated[11].

2.2 Chemical Transport Models for PM_{2.5} Predictions

Chemical transport models (CTMs) consist of the differential equations of the relevant physical and chemical atmospheric processes. In other words, CTMs are prognostic models that, given the emission rates of selected pollutants and their precursors and prevailing meteorological conditions, use numerical algorithms to predict the pollutant concentrations based on a combination of fundamental and empirical representations of the relevant physicochemical atmospheric processes[13]. Some processes are required to be simulated in any CTM, which include [15]:

- Emissions
- Meteorology
- Transport and diffusion processes
- Chemical transformations
- Representation of PM
- Deposition processes

In conclusion, the general formulation of the atmospheric processes that need to be simulated in any CTM is presented in Figure 2.2[13]. Even though the major physicochemical processes treated in most CTMs are identical, how to characterize the chemical composition and size distribution of PM in different CTMs is varied. Nowadays, many CTMs have been developed and some widely used open-source systems include EMEP, WRF-CHEM, CMAQ, and Lotos-Euros[14]. In this review, a brief introduction to Lotos-Euros and CMAQ will be made in section 2.1 and 2.2 respectively.

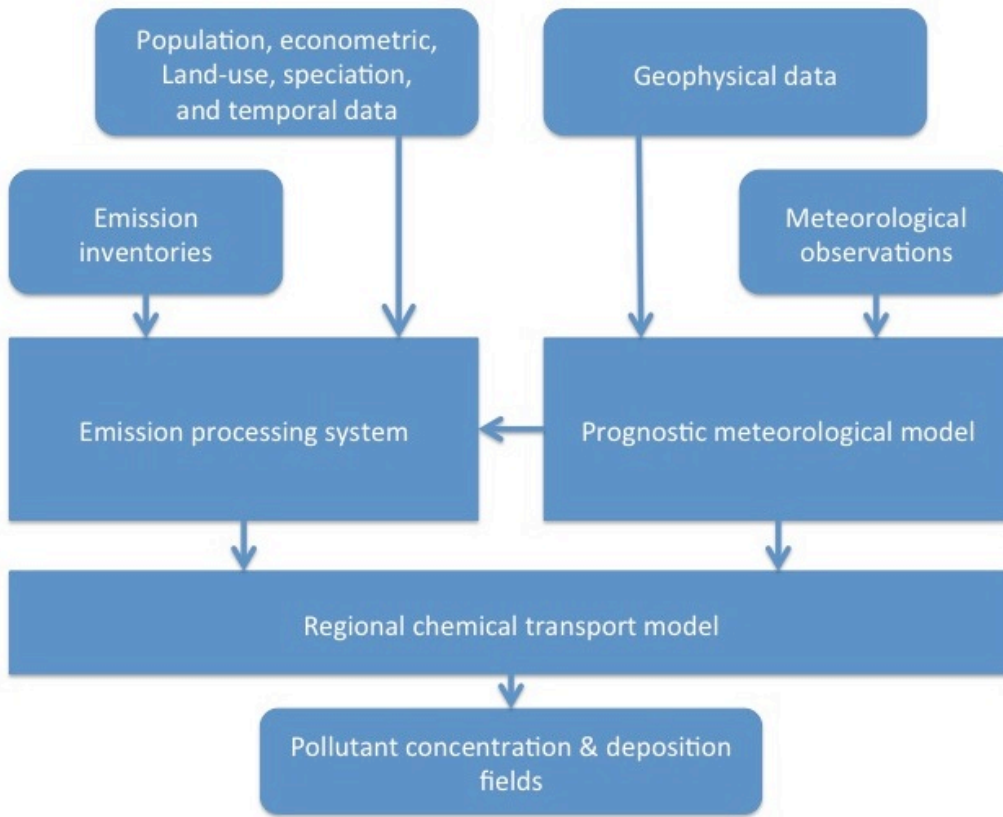


Figure 2.2. The components of a PM modeling system (redraw based on[13])

2.2.1 Lotos-Euros

Lotos-Euros as a three-dimensional CTMs-based system is originally designed for assessing air pollution over Europe. The main prognostic equation treated in Lotos-Euros is [15]:

$$\frac{\partial C}{\partial t} + U \frac{\partial C}{\partial x} + V \frac{\partial C}{\partial y} + W \frac{\partial C}{\partial z} = \frac{\partial}{\partial x} \left(K_h \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_h \frac{\partial C}{\partial y} \right) + \frac{\partial}{\partial z} \left(K_z \frac{\partial C}{\partial z} \right) + E + R + Q - D - H$$

with C the concentration of a pollutant, U , V and W being the large-scale wind components in respectively west-east direction, in south-north direction and in vertical direction. K_h and K_z are the horizontal and vertical turbulent diffusion coefficients. E represents the entrainment or detrainment due to variations in layer

height. R gives the amount of material produced or destroyed as a result of chemistry. Q is the contribution by emissions, and D and H are loss terms due to processes of dry and wet deposition respectively. In Lotos-Euros, operator splitting is used to solve this equation numerically.

R. Timmermans et al. (2017) [16] applied Lotos-Euros in China to track how the predefined source sectors contributed. They pointed out that a systematic underestimation of PM concentrations in Beijing using Lotos-Euros existed and analyzed the reasons for it. The summary of the mentioned reasons is shown in Table 2.1. Some components were ignored but indeed contributed to particulate matters while other components were taken into account but underestimated.

Table 2.1 Reasons for the systematic underestimation of Lotos-Euros

Missing Components	Underestimated Components
road dust resuspension, fugitive dust from deserts, construction works, demolition works, secondary organic aerosol(SOA), heterogeneous sulphate,...	all aerosol components(including sulphate, organic matter),...

2.2.2 CMAQ

The Community Multiscale Air Quality Modeling (CMAQ) System (www.epa.gov/cmaq) is another CTMs-based method for conducting air quality model simulations and as an active open-source project of the United States Environmental Protection Agency (U.S. EPA). CMAQ was used to simulate the PM_{2.5} formation and its response to precursor emission reductions in California's San Joaquin Valley (SJV) (J Chen et al., 2014[18]). Prank M et al. (2016) [19] evaluated the performance of four CTMs (Lotos-Euros, CMAQ, EMEP, SILAM) in forecasting the aerosol chemical composition over Europe. The results showed that all these four models systematically underestimated PM₁₀ and PM_{2.5} by 10–60%, depending on the model and the season of the year.

2.3 Traditional Statistical Methods for PM2.5 Predictions

Autoregressive Integrated Moving Average (ARIMA) and Multiple Linear Regression (MLR) have been used to predict air pollutant concentrations in many regions. However, because of their linear representation of non-linear systems, the predictions of these two methods are of variable accuracy[21]. Other commonly used methods include Support Vector Regression (SVR) (Osowski S et al., 2007[22]), Kalman Filter (KF) (Hoi et al., 2008[23]) and Hidden Markov Model (HMM) (Sun W et al., 2013[24]).

In addition to the above approaches, Neural Networks (NN), which can perform nonlinear mapping, generally provide superior performance for complex systems. Therefore, it has been widely used for forecasting tasks. What's more, in order to improve the network performance in specific tasks, many variants of neural networks have been put forward. For more detailed description regarding neural networks, see section 2.4.

2.4 Deep Learning Methods for PM2.5 Predictions

Deep learning as a fresh and promising branch of machine learning has received immense attention in both academy and industry. It has been successfully applied to image classification, natural language processing, prediction tasks, object detection and so on[6]. With the usage of multiple layer architectures to extract inherent features layer by layer, deep learning algorithms are able to recognize the representative characteristics within data. That is, for air pollutant predictions, deep learning can be powerful in extracting representative air quality features without prior knowledge, which is likely to result in superior prediction performance.

2.4.1 Autoencoders

Autoencoders as a particular form of deep feedforward networks can learn the features from unlabeled data in an unsupervised way. For an autoencoder, its output is set to be equivalent to its input, after which the network is trained to minimize the error between them[25]. Nowadays, many kinds of research have been done by using autoencoders as pre-training methods to extract representative spatiotemporal features.

Li X et al. (2016) [10] developed a spatiotemporal deep learning (STDL) based air quality prediction method. A stacked autoencoder was used to extract representative features. In addition to autoencoder layers, a logistic regression (LR) layer was added at the top for the purpose of real-value predictions. It was demonstrated by experimental results that the proposed method could have better performance than autoregressive moving average (ARMA) and support vector regression (SVR).

Bun Theang Ong et al. (2016) [25] predicted PM_{2.5} concentrations in 52 cities within Japan. A deep recurrent neural network (DRNN) was proposed with a novel pre-training method (DynPT) using a specially designed autoencoder. The experimental results showed that the proposed DRNN outperformed the PM_{2.5} prediction system VENUS, which is based on CTMs.

2.4.2 Deep Belief Networks

Deep belief networks (DBNs) were one of the first non-convolutional networks to successfully admit training of deep architectures. With its introduction in 2006, the current deep learning renaissance began[27]. Several restricted Boltzmann machine (RBM) layers adding one back-propagation (BP) layer constitute a DBN, which is applicable for both classification and prediction problems.

A geo-intelligent DBN (Geoi-DBN), which incorporates geographical correlation including geographical distance and spatiotemporally correlated PM2.5 into deep learning, was used to estimate ground-level PM2.5 (Li T et al., 2017[28]). It was showed that the Geoi-DBN can capture the essential features associated with PM2.5 from the latent factors and perform significantly better than traditional neural networks including back propagation neural networks (BPNN) and generalized regression neural networks (GRNN).

2.4.3 Recurrent Neural Networks

Recurrent neural networks (RNN) as a special class of neural networks pose feedback connections between units, which lead to directed cycles. These connections allow RNN to exhibit dynamic temporal behavior by combining information in their past inputs to compute future outputs [25]. The hidden states and nonlinear behaviors make RNN particularly suitable for integrating the information over many time steps and for explaining complex sequential relationships. More detailed description of recurrent networks will be made in section 3.1.

Fabio Biancofiore et al. (2017) [29] used meteorological information and PM10 concentrations as input to forecast the daily averaged PM10 concentrations. All experiments showed that the neural network with recursive architectures had better performance compared to both MLR and the neural network without recursive connections.

2.4.4 Long Short-Term Memory Neural Networks

The basic problem of learning long-term dependencies in RNN is that gradients propagated over many stages tend to either vanish or explode. Even though suitable methods are adopted to ensure the stability of RNN, the difficulty within long-term

dependencies arises from the exponentially smaller weights given to long-term interactions compared to short-term ones[27]. The causes of this phenomenon will be further explained in section 3.1.

To solve these issues, LSTM NN as a special kind of RNN were developed by Hochreiter and Schmidhuber[30]. The basic structure of LSTM NN and RNN is similar, but the neurons of LSTM NN are replaced by memory blocks. LSTM NN introduce gate mechanism to prevent gradients from vanishing or exploding and are capable of learning long time series. The formulation of an LSTM block will be introduced in section 3.2.

Fan J et al. (2017) [9] proposed a spatiotemporal prediction framework that consists of LSTM layers for PM_{2.5} predictions. Experiments showed that the proposed framework outperformed both deep feedforward neural networks (DFNN) and gradient boosting decision trees (GBDT). Xiang Li et al. (2017) [6] used a novel long short-term memory neural network extended (LSTME) model that inherently considered spatiotemporal correlations for air pollutant predictions. The experiments were compared with many data based methods including ARMA, SVR, traditional LSTM NN and time delay neural networks (TDNN). The results demonstrated its superior performance compared to all of them. Reddy V et al. (2018)[2] used an LSTM based seq2seq model to forecast air pollution in Beijing, China. The results showed that the LSTM framework produced equivalent accuracy when predicting future sequences compared to using SVR for a single timestamp.

In addition to air quality, many kinds of research for the predictive learning in weather and atmospheres have been done based on LSTM NN. Ghaderi A et al. (2017) [31] presented a spatiotemporal wind speed forecasting algorithm using deep learning and in particular, LSTM NN. Zhang Q et al. (2017) [32] adopted an LSTM-based network to predict sea surface temperature (SST). Zaytar M A et al. (2016)[33] forecasted weather data including temperature, humidity and wind speed 24 and 72 hours in advance with the usage of LSTM NN.

2.4.5 Hybrid Models

Recently, many researchers have combined neural networks with other techniques such as traditional statistical methods and wavelet transformation so as to improve the forecasting accuracy. Some examples are briefly described as follows.

Díaz-Robles et al. (2008) [21] proposed a novel hybrid model that combines ARIMA and Artificial Neural Networks (ANN) to improve the accuracy of PM10 predictions in Temuco, Chile. Experimental results showed that the hybrid model could effectively improve the forecasting accuracy compared to either of the models that used separately. Feng X et al. (2015)[1] presented a hybrid model combining air mass trajectory analysis and wavelet transformation to improve the ANN forecast accuracy of daily average concentrations of PM2.5. It was demonstrated that the trajectory-based geographic models and wavelet transformation were effective tools for model improvement. Prakash A et al. (2011)[34] proposed a wavelet-based recurrent neural network to forecast concentrations of ambient CO, NO₂, NO, O₃, SO₂, and PM2.5. According to the experiments, with a judicious selection of wavelet network, prediction accuracy could be increased significantly.

2.5 Related Work on Visibility Predictions

As mentioned in section 1.2, a mature physical or mathematical model for visibility predictions does not exist. Up to now, many kinds of research have been carried out on visibility through observations, models, climatology analysis and statistical methods[35]. For investigating the relationship between visibility and other features, Fan G et al. (2016) [36] revealed that relative humidity and PM2.5 concentrations were closely related to visibility. The increase of relative humidity and PM2.5 concentrations can lead to the decline of visibility. Wang X et al. (2016) [37] pointed out that a power function relation existed between PM2.5 and visibility while an exponential relation existed between relative humidity and visibility.

As for the prediction of visibility, numerical weather prediction (NWP)

approaches are normally considered as large-scale or mesoscale systems while statistical methods are commonly used in small-scale problems.

2.5.1 Numerical Weather Prediction

The motion of viscous fluid substances in the atmosphere is governed by a set of equations, known as the *Navier-Stokes* equations. NWP approaches solve these equations numerically by computers and then produce weather forecasts including visibility predictions. The most common procedure is to analyze simulation data of three-dimensional global models, such as the Global Forecast System (GFS) from the National Oceanic and Atmospheric Administration (NOAA), the Integrated Forecasting System (IFS) from the European Centre for Medium-Range Weather Forecasts (ECMWF), or mesoscale models, such as the Weather Research and Forecasting (WRF) model[8].

However, sometimes, the forecasts of poor-visibility events by NWP are restricted due to the fact that the low-visibility event relies heavily on small-scale variations of the atmosphere while extremely high resolutions are required to accurately simulate these variations. Hence, statistical methods are commonly used to solve small-scale problems.

2.5.2 Statistical Methods

Dutta D et al. (2015) [35] identified the key environmental and meteorological parameters for visibility predictions through decision trees and the selected parameters were NO₂, wind speed, relative humidity, CO, and temperature. After selecting input features, a multi-layer perceptron (MLP) was used to forecast visibility 6 hours in advance during winter at Kolkata airport, India. Zhu L et al. (2017) [4] also used a MLP to predict the dominant visibility 1 hour in advance at Urumqi Airport, China. They pointed out that the absolute error of the hourly

prediction was 706 m. When the visibility was smaller than 1000 m, the prediction error was 325 m.

Fuzzy time series combining ARIMA was adopted by Yao J et al. (2013)[7] to predict the atmospheric visibility in Shanghai, China and the relative error between the model outputs and observations was acceptable in practice. Cornejo-Bueno L et al. (2017)[8] used SVR, extreme-learning machines (ELM) and Gaussian-process algorithms to efficiently predict low-visibility events at Valladolid airport, Spain. Among them, the Gaussian process showed the best performance.

2.6 Summary

The aim of this chapter is to review the literature on both PM2.5 and visibility predictions. According to section 2.4.4, many kinds of research have been done to show the powerful performance of LSTM NN in both PM2.5 and weather predictions. What's more, at the beginning of section 2.5, it was mentioned that many researchers revealed that there is a strong relationship between PM2.5 and visibility. However, as illustrated in section 2.5.2, up to now, we still have not found any literature involves using LSTM NN to predict visibility. *To summarize, LSTM NN have been applied for PM2.5 predictions successfully and it has been revealed that there is a strong relationship between PM2.5 and visibility. However, up to now, LSTM NN have not been tried for visibility predictions.* Since our goal is to establish deep learning architectures that can forecast PM2.5 and visibility accurately, according to the above analysis, LSTM NN are extremely appropriate for achieving this objective.

As illustrated at the beginning of this chapter, in this research, we deal with ST data. Two special properties of ST data are auto-correlation and heterogeneity existing in both time and space. Our research leverages information about spatial neighborhoods to enforce spatial consistency.

The literature reviews of PM2.5 predictions were divided into three parts: CTMs-based systems, traditional statistical methods and deep learning approaches. Famous CTMs-based open-source systems include Lotos-Euros CMAQ, EMEP, and

WRF-CHEM. As for traditional statistical methods, ARIMA, MLR, SVM, KF, and HMM have been tried by many researchers. When it comes to deep learning approaches, autoencoders, DBN, RNN, and LSTM NN have been widely used for air pollutant predictions. Some hybrid models combining neural networks with other techniques were also described in this review.

In terms of visibility predictions, current methods include NWP and statistical methods. NWP models such as GFS, IFS, and WRF are commonly used. As for statistical methods, ARIMA, SVR, ELM, Gaussian-process, and MLP have been implemented by other researchers for visibility tasks.

3. Theoretical Background for Deep Learning

Algorithms

In the following chapter 4 and 5, deep learning architectures for PM2.5 and visibility predictions will be built. Some theoretical backgrounds such as how the LSTM block looks like, how to minimize the target function, how to avoid overfitting during training and how to evaluate the framework performance will be introduced in this chapter.

3.1 Recurrent Neural Networks

Recurrent neural networks (RNN) as a family of neural networks are very suitable for processing sequential data. Compared with feedforward neural networks (DFNN), essentially any function involving recurrence can be considered as a recurrent neural network[27]. Hence, RNN can have many variants. One of the important RNN defines the hidden units as Eq 3.1.

$$h_t = f(h_{t-1}, x_t; \theta) \quad (3.1)$$

Here, $x_t \in R^K$ and K is the input dimension. With this recurrence, an arbitrary length sequence $(x_t, x_{t-1}, \dots, x_2, x_1)$ is mapped into a fixed length vector h_t .

When the recurrent network is trained to perform a task that predicts the future according to the past, the network typically learns to use h_t as a kind of lossy summary of the task-relevant aspects of the past sequence of inputs up to t [27]. Fig 4.1 shows a recurrent network that is based on Eq 3.1. It consists of one input layer, one output layer, and one recurrently connected hidden layer. *Notably, each node in the graph is a vector representing a layer.* In details, the model input is $x = (x_1, x_2, \dots, x_T)$, where $x_i \in R^K, i = 1, 2, \dots, T$; K is the input dimension; T represents the input time lag; and the model output is $y = (y_1, y_2, \dots, y_T)$. The recurrent network processes information from the input sequence x by incorporating it into the state vector h that is updated over time. And the corresponding output sequence is the

vector y .

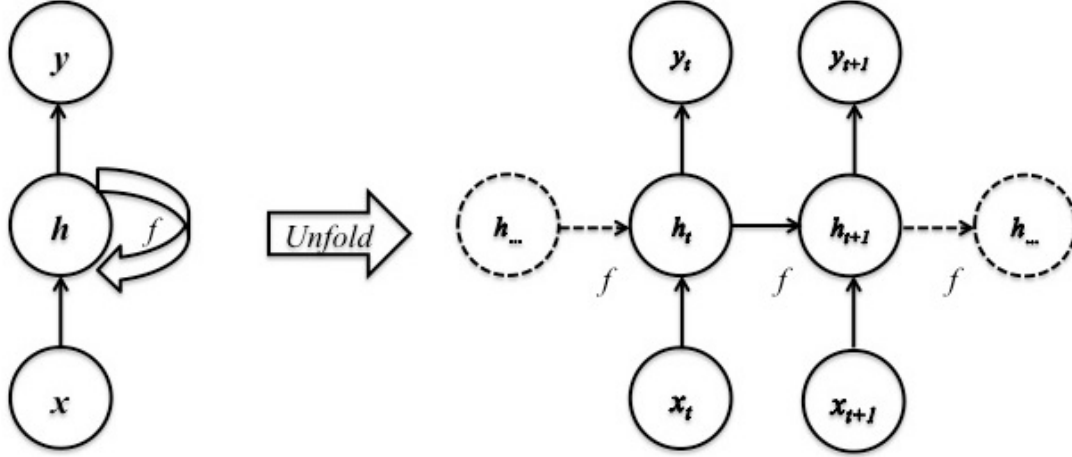


Figure 3.1 (Left) A circuit diagram. (Right) The same network in an unfolded form. (redraw based on [27])

The mathematical problem of learning long-term dependencies in RNN is that gradients propagate over many stages tend to either vanish or explode. As shown in Fig 3.1, the function f is involved in each timestamp. If f is considered as a linear function and the input x_t is ignored, then Eq 3.1 can be rewritten as

$$h_t = W^T h_{t-1}$$

This recurrence relation essentially describes the power relationship and can be simplified as

$$\begin{aligned} h_t &= (W^T)^t h_0 \\ &= (W^t)^T h_0 \end{aligned}$$

If W admits the eigendecomposition, then

$$W^t = (Q \text{diag}(\lambda) Q^{-1})^t = Q \text{diag}(\lambda)^t Q^{-1}$$

Therefore, any eigenvalue λ_i with magnitude less than one will decay to zero and eigenvalue with magnitude greater than one tends to explode. *The gradient vanishing and exploding problem mentioned above refers to the fact that gradients through such a process are also scaled according to $\text{diag}(\lambda)^t$.* Since gradient descent is the most popular algorithm to optimize the network parameters, vanishing gradients

make it difficult to determine which direction the parameters should move to reduce the cost function, while exploding gradients can make learning unstable.

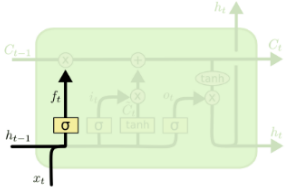
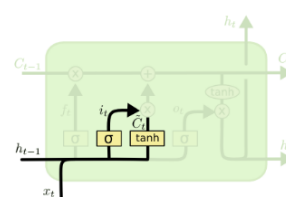
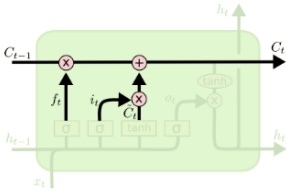
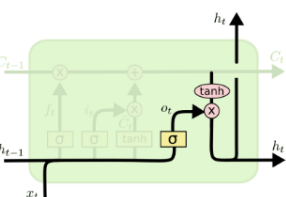
Various techniques have been developed to overcome the above difficulties. The gradient explosion problem is usually solved by gradient clipping. That is, the gradients are clipped to a threshold value to prevent them from getting too large. As for gradient vanishing, the most effective and common methods used in practice are gated RNN. These gated units include the LSTM unit and gated recurrent unit (GRU)[27]. In the next section, we will make a brief description of LSTM NN.

3.2 LSTM NN Formulation

LSTM NN as a special type of RNN, in addition to self-recurrent memory cells, introduce three units (input, output and forget gates). That is, instead of simply applying an element-wise affine transformation of inputs and recurrent units like Eq 3.1, LSTM NN use “LSTM cells” through these three gates to have an internal recurrence. The LSTM memory block can have many variants and these variants usually differ in the input for the memory. In our research, the LSTM memory block within a single cell is given in Table 3.1[30]. $[h_{t-1}, x_t]$ is the input for all gates.

In this Table, i_t , o_t , and f_t correspond to the activation of the input gate, output gate and forget gate, respectively; C_t and h_t are the activation for each cell and memory block respectively; and W and b represent the corresponding weight matrix and bias vector respectively. *With the introduction of three gates, adding the self-recurrent memory, there are four weight matrices (W_f , W_i , W_c and W_o) in each LSTM cell. So the number of parameters for an LSTM neural network increases at least four times compared to an origin feedforward network, in which only one weight matrix exists.*

Table 3.1 From *Colah blog*: The Forward Training Process of the LSTM NN

	Formula	Usage
Forget Gate 	$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$	Determine what information is thrown away from the cell state.
Input Gate 	$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$	Determine what new information is stored in the cell state.
Update 	$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$	Update the old cell state C_{t-1} , into the new cell state C_t .
Output Gate 	$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$ $h_t = o_t * \tanh(C_t)$	Determine what information is contained in the output

3.3 Dropout in Recurrent Neural Networks

Deep neural networks with numerous parameters are indeed very powerful machine learning methods. However, such networks easily face the problem of overfitting. As discussed at the end of section 3.2, the number of weights for an LSTM neural network increases at least four times, which makes it encounter overfitting much more easily.

Srivastava N et al. (2013)[38] put forward dropout, which has been considered

as the most powerful regularization method for DFNN to avoid overfitting. Its key idea is to randomly drop units along with their connections from the network during training.

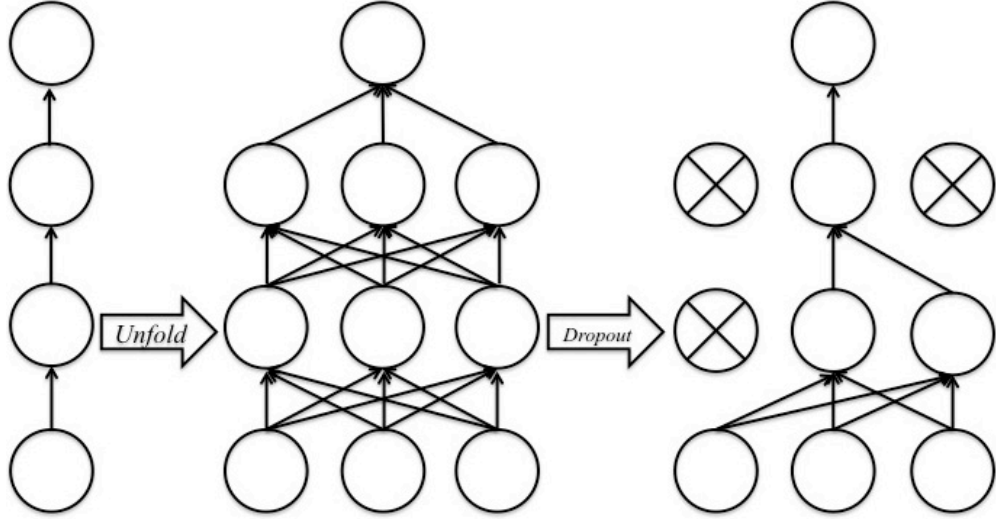


Figure 3.2 Dropout in a neural network. (Left) Each node in the graph represents a layer. (Middle) Every neuron is a node in the graph. (Right) An example of a thinned network produced by applying dropout to the network in the middle. Crossed units have been dropped. (redraw based on [38])

In order to successfully apply dropout in RNN, Gal Y et al. (2016)[39] proposed a new variational inference based dropout technique for the LSTM. The forward training process of the LSTM given in Table 3.1 can be rewritten as:

$$\begin{pmatrix} f_t \\ i_t \\ \tilde{C}_t \\ o_t \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{tanh} \\ \text{sigm} \end{pmatrix} \left(\begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \cdot W \right) \quad (3.2)$$

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

$$h_t = o_t * \tanh(C_t)$$

Following[39], implementing approximate inference equals implementing dropout in RNN with the same network units dropped at each time step, randomly dropping inputs, outputs, and recurrent connections. By means of dropout, the above parameterization (Eq 3.2) can be rewritten as

$$\begin{pmatrix} f_t \\ i_t \\ g_t \\ o_t \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} \left(\begin{pmatrix} x_t * z_x \\ h_{t-1} * z_h \end{pmatrix} \cdot W \right)$$

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

$$h_t = o_t * \text{tanh}(C_t)$$

with z_x and z_h represent random masks repeated at all time steps.

Figure 3.3 shows how to apply this kind of dropout in a recurrent neural network. The network contains one input layer, two hidden layers, and one output layer. Each circle represents a neuron, with horizontal arrows corresponding to recurrent connections. Besides, vertical arrows represent the input and output of each cell. *Colored connections indicate dropped input and different dropout masks are represented by various colors. Notably, the identical dropout mask is used at each time step, including the recurrent layers.*

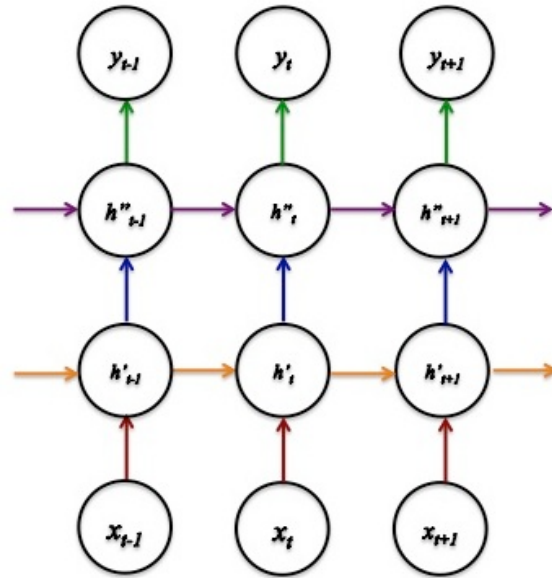


Figure 3.3 Dropout in a recurrent neural network (redraw based on [39]).

3.4 Gradient Descent Optimization Algorithms

Gradient descent as a popular optimization algorithm minimizes an objective function by updating the parameters in the opposite direction of the gradient. The learning rate η determines the size of the steps used during an update. The iteration formula can be described as:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta)$$

Since numerous neurons exist in neural networks, and the weights between neurons in different layers are the parameters we need to calculate, gradient descent as a first-order method is feasible to compute in practice and commonly used to optimize neural networks.

3.4.1 Mini-batch Gradient Descent

For neural networks, gradient descent methods can be divided into three types and the difference lies in how many data included in the objective function. Batch gradient descent (BGD) i.e. vanilla gradient descent computes the gradient of the cost function w.r.t. the parameters θ for the entire training dataset $\sum_{i=1}^N (x^{(i)}, y^{(i)})$, that is

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(1:N)}, y^{(1:N)})$$

Before each parameter update, we need to calculate the gradient of each sample in the dataset and then average it or sum it up, so batch gradient descent can be very slow sometimes.

In contrast, stochastic gradient descent (SGD) computes the gradient of a single training example $(x^{(i)}, y^{(i)})$ only:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}, y^{(i)})$$

Because SGD performs update according to just one sample, it is usually much faster.

However, the selection of training samples is of high variance in this method and it sometimes causes the objective function to fluctuate heavily.

In order to make a trade-off between the accuracy of an update (BGD) and the time consumed of an update (SGD), mini-batch gradient descent are popular in optimizing neural networks, in which an update is performed according to mini-batch n ($n < N$) training examples:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(ii+n)}, y^{(ii+n)})$$

However, good convergence cannot be guaranteed in such a mini-batch gradient descent algorithm and several problems are required to be solved[40]:

- How to choose a proper learning rate.
- How to adjust learning rates for different parameter updates.
- Since the target function of the neural network is non-convex, how to avoid getting trapped in their numerous suboptimal local minima.

3.4.2 Adam

Many algorithms have been developed by the Deep Learning community to address the problems mentioned above. Adaptive Moment Estimation (Adam) (Kingma D P et al. (2014)[41]) is a commonly used method that is able to compute adaptive learning rates for each parameter at every time step t . This method has many benefits. For example, it is straightforward for implementation, is efficient in terms of computation, requires little memory, and is suitable for problems with numerous data or parameters[41].

There are two key parameters in Adam. v_t keeps an exponentially decaying average of past squared gradients while m_t stores an exponentially decaying average of past gradients.

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned}$$

Here, m_t and v_t estimates the first moment (the mean) and the second moment (the

uncentered variance) of the gradients respectively. The term m_t increases for dimensions whose gradients point in the same directions and reduces updates for dimensions whose gradients change directions. By introducing v_t , Adam adjusts the learning rates according to the parameters, larger updates are performed for small gradients while smaller updates are implemented for large gradients.

However, if these moving averages are initialized as vectors of zeros, the moment estimates will be biased towards zero, especially during the initial timesteps, and when the decay rates are small[41]. The bias-corrected m_t and v_t are given as:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Here, β_1^t and β_2^t means β_1 and β_2 to the power t respectively. Then, the Adam update rule is given as

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

The authors proposed default values of 0.9 for β_1 , 0.999 for β_2 , and 10^{-8} for ϵ .

An important property of Adam lies in its choice of learning rate η . It was proved that the effective magnitude of the steps taken in parameter space at each timestep is approximately bounded by the stepsize η , that is[41]

$$|\Delta_t| \lesssim \eta$$

According to this property, the right magnitude of η can be determined beforehand.

3.5 Performance Measures

Because PM2.5 concentrations are predicted as a regression task while visibility forecasts are considered as a classification task, the evaluation metrics for these two tasks will be mentioned separately in the following two sections.

3.5.1 Regression Task

Since the PM2.5 prediction is considered as a kind of regression tasks, root mean square errors (RMSE) and mean absolute errors (MAE) will be used to measure the overall closeness of the results with the reference values. The formulations of these two indicators are as follows.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_i^*)^2}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_i^*|$$

where y_i^* is the observed concentration, y_i is the predicted value and n refers to the total number of testing samples. In PM2.5 predictions, the mean squared error (MSE), which is the square of RMSE, will be selected as the target function that needs to be minimized.

However, RMSE and MAE just show the average performance of the prediction. Considering the variability properties of PM2.5 data and the severe influence of high PM2.5 concentrations, the regression results will be classified into the corresponding air quality level according to the Chinese Technical Regulation on the Ambient Air Quality Index (see Table 3.2)[6].

Table 3.2 PM2.5 air quality levels

Rank	Range($\mu\text{g} / \text{m}^3$)	Description
L1	<35	Good
L2	(35,75)	Moderate
L3	(75,115)	Unhealthy for sensitive groups
L4	(115,150)	Unhealthy
L5	(150,250)	Very unhealthy
L6	>250	Hazardous

After the conversion, measures such as the recall rate J_{r_i} and precision rate J_{p_i} can be utilized. These measures are evaluated per class i as

$$J_{r-i} = \frac{tp_i}{tp_i + fn_i}$$

$$J_{p-i} = \frac{tp_i}{tp_i + fp_i}$$

where tp_i corresponds to the number of true positive predictions (the sample within the class i is predicted as the class i), fp_i represents the number of false positive predictions (the sample within other classes is predicted as the class i), and fn_i indicates the number of false negative predictions (the sample within the class i is predicted as other classes).

These symbols are summarized in Table 3.3, in which a confusion matrix for a binary classifier is shown. Here, True and False can be interpreted as the two classes of the binary classifier respectively. This matrix can be extended to the case of more than two classes easily. For a confusion matrix, labels in the row direction show the measured data while labels in the column direction show the predicted data.

Table 3.3 A Confusion Matrix

	Predicted True	Predicted False
Actual True	tp	fn
Actual False	fp	tn

Afterward, the total accuracy rate J_a is formulated as

$$J_a = \frac{tp + tn}{tp + tn + fn + fp}$$

3.5.2 Classification Task

For visibility predictions, it is considered as a classification task. The reason why we convert it into a classification problem will be described in section 5.3. In a classification problem, numbers cannot be used to represent classes immediately because the difference between these numerical values cannot reflect the difference and relationship between different class correctly. Instead, 1-of-N encoding should

be used for the target y . So if input x belongs to class C_k , then y is a binary vector of length N (the number of classes) containing a single 1 for element k (the correct class) and 0 elsewhere.

According to the Airport Operation Regularization published by the Civil Aviation Administration of China (CAAC) (www.caac.gov.cn), the minimum visibility ranges for different types of aircrafts to take off are given in Table 3.4.

Table 3.4 Visibility standards for aircrafts to take off

Rank	Range (m)	Description
L1	(0,800)	For aircrafts capable of low visibility flying
L2	(800,1600)	For three-engine aircrafts and four-engine aircrafts
L3	>1600	For all planes

Because there are three different classes in our task, then, an input belonging to L2 will be given a target vector:

$$y = (0, 1, 0)^T$$

For visibility predictions, an appropriate activation function, such as softmax function, will be selected in the final layer so that the output in each neuron is mapped into a value between 0 and 1 and the sum of all output neurons is 1. With the help of 1-of-N encoding, the neural network is trained to have a single high-activation output when a certain input is present. It can be interpreted as the probability that it belongs to each class.

In order to better minimize the difference between these two probability distributions, cross-entropy will be used as the target function. Additionally, since there is the ‘*class-imbalance*’ phenomenon in the visibility dataset, which will be described more specifically in section 5.3.1, weighted cross-entropy will be employed instead. It is defined as

$$C = - \sum_{n=1}^N \left[\alpha_k \sum_{k=1}^K t_{nk} \ln y_{nk} \right]$$

Here, N refers to the batch size, K refers to the number of classes, α_k is the corresponding weight for class k , t_{nk} is the 1-of-N encoding of the target while y_{nk} is the neural network output.

As for the performance evaluation in this classification task, the recall rate, precision rate, accuracy rate and confusion matrix described in section 3.5.1 will be adopted.

3.6 Summary

In this chapter, we gave a brief introduction to the theoretical background regarding deep learning. In section 3.1, RNN were described and the reason why they easily encounter the problem of gradient vanishing or exploding was analyzed. As a solution to this problem, LSTM NN were introduced in section 3.2. However, since LSTM NN introduce gate mechanism, the number of parameters increases at least four times compared to DFNN, which makes LSTM NN encounter the problem of overfitting much more easily. Then, dropout as a regularization technique was described in section 3.3. One of the most challenging problems of neural networks is how to minimize the target function, which leads to the introduction of optimization algorithms. In section 3.4, the mini batch gradient descent formula and the learning rate update method Adam were introduced. As for section 3.5, we made a brief description of the target function and performance indicators for regression and prediction task respectively. The cost function of PM2.5 predictions will be MSE while the target function of visibility predictions will be weighted cross-entropy. The recall rate, precision rate, and confusion matrix will be used to evaluate performance in both regression task and classification task.

4. Methodology for PM2.5 and Visibility Predictions

Using Deep Learning

At the beginning of this chapter, the research problem will be described in details. Afterward, data visualization and statistical analysis will be implemented in the two corresponding datasets. The data preprocessing method for air quality features with high missing rates will be discussed in section 4.5.

4.1 Problem Description of PM2.5 Predictions

As described in section 2.2, PM2.5 concentrations are related to the relevant physical and chemical atmospheric processes. Historical information including air quality and meteorological data was collected as input for the proposed network. Given Beijing's geographical location, northerly wind is very beneficial for the dispersion of particulates, whereas southerly wind may sometimes aggregate pollution[42].

Considering the importance of wind direction and the strong spatial correlations among these stations (shown in section 4.3.3), not only the meteorological features such as wind direction and wind speed at the target station, but also the PM2.5 concentrations at three geographically nearest monitoring stations will be taken as input. To summarize, for the short-term forecast in a single station, data of all selected features of this station will be injected, along with PM2.5 data of three surrounding stations. Within this framework, spatial and temporal correlations are represented by neighboring stations and the 'memory' of LSTM respectively.

In the real world, PM2.5 concentrations should be predicted many hours in advance so that there is enough time for warning and taking measures. However, with the increase of the prediction interval, the predicting task becomes much more difficult and the performance tends to degrade [2]. There is a trade-off between high accuracy and a long prediction period. *In our proposed LSTM methodology, the goal is to forecast PM2.5 12 hours ahead.*

To summarize, for a single station n , the input dataset can be denoted as

$$st_n = \{x_1, x_2, \dots, x_T\}$$

T refers to the sequential length. If the number of selected features is k , then each observation x_t is a $k + 3$ dimensional vector because PM2.5 concentrations at the three geographically nearest monitoring stations are considered. The network output is the prediction of this single station 12 hours ahead.

4.2 Problem Description of Visibility Predictions

One obvious difference between PM2.5 and visibility predictions is that PM2.5 data is obtained from air quality monitoring stations while visibility data is recorded in metrological stations. Due to the limitation of accessibility, meteorological data was accessed at just one station, and information at nearby stations was not taken into account in this visibility task.

As illustrated in section 1.2, visibility is mainly influenced by airborne particles and weather patterns. Therefore, air quality and meteorological information will be used as input for visibility predictions. *Here, the prediction period is fixed to be 4 hours, which indeed has practical meanings because airport operators can have enough time to take measures to reduce the economic loss and passengers inconvenience caused by the air traffic disruption of low visibility.*

In conclusion, for the target airport, the input dataset can be denoted as

$$st = \{x_1, x_2, \dots, x_T\}$$

T refers to the sequential length and each observation x_t is a k dimensional vector in which k refers to the number of selected features. The network output is the predicted visibility level of this airport 4 hours ahead.

4.3 Dataset for PM2.5 predictions

For PM2.5 forecasts, the hourly air quality data for Beijing City from 2013/01/18 16:00 to 2016/10/31 23:00 at 11 air quality monitoring stations and meteorological data every half hour from the same period at Beijing Capital International Airport were downloaded from the Qingyue Open Environmental Data Center

(<https://data.epmap.org>). Figure 4.1 shows the distribution of these air quality monitoring stations and the meteorological station.

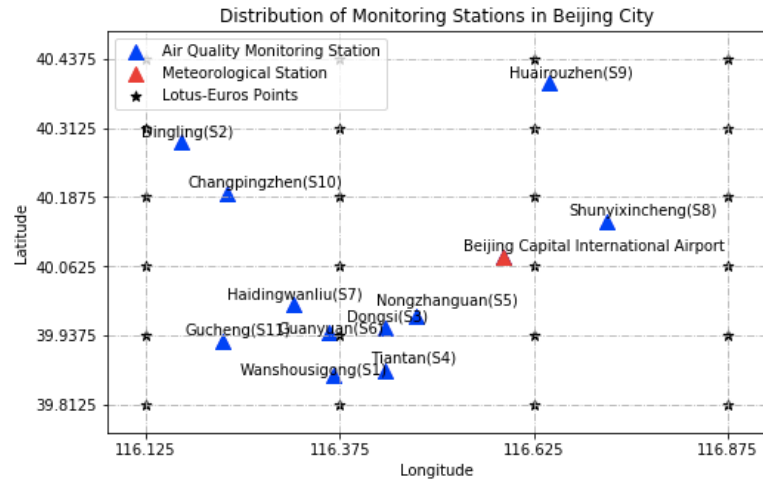


Figure 4.1 The distribution of monitoring stations in Beijing

4.3.1 Meteorological Data Visualization

Even though the meteorological data was obtained every half hour, it should be consistent with the air quality data, which indicates that the final meteorological data is hourly. The number of valid data instances at integral time points and half time points are given in Table 4.1. According to Table 4.1, in terms of available data size, there is no significant difference between these two datasets. However, this just shows the comparison quantitatively, the distributions of missing parts for different features are investigated in Figure 4.2.

Table 4.1 Valid data instances for different meteorological features

Feature	Origin	minute =0	minute =30	Total Missing Rate
Wind Direction	51273	25683	25590	22.73%
Wind Speed	66001	33011	32990	5.29%
Temperature	65995	33009	32986	5.38%
Dew Point	65966	32991	32975	5.82%
Visibility	36587	21015	17772	44.86%

Figure 4.2 gives the dataset visualization results, in which white parts correspond to missing parts. For the feature wind direction, the missing parts are distributed throughout the whole sequence. Besides, it is noticeable that the missing phenomenon of visibility is rather severe. We will discuss whether to contain this feature as an input feature through experiments in section 5.1.2.

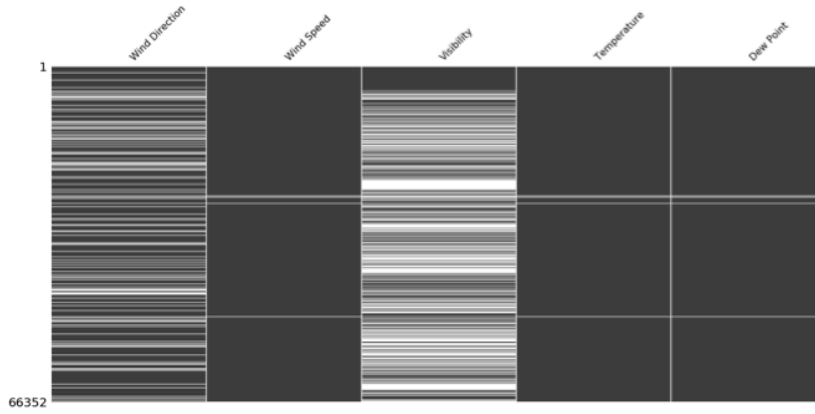


Figure 4.2 The visualization of the meteorological data

Based on Table 4.1 and Figure 4.2, the number of available data instances at integral time points and half time points do not show obvious difference while the missing parts do not show significant continuous characteristics. Linear interpolation was done in the whole dataset and then the values corresponding to integral time points were selected. The statistical summary of the measured meteorological features is shown in Table 4.2.

Table 4.2 The statistical summary of the meteorological data

Variable	Unit	Range	Mean	Std
Temperature	°C	[-16,42]	13.93	11.63
Wind direction	(°)	[10,360]	175.02	111.18
Wind speed	<i>m/s</i>	[0,20]	2.89	2.17
Visibility	<i>m</i>	[0,30000]	11611.17	8802.04
Dew point	°C	[-40,27]	3.04	13.90

4.3.2 Air Quality Data Visualization

Since datasets corresponding to different monitoring stations are quite similar, station S6 (Guanyuan, Beijing) will be used as an example dataset to show air quality data. The statistical summary of the measured air quality variables is given in Table 4.3 while Figure 4.3 shows the data visualization results.

Table 4.3 The statistical summary of the air quality data

Variable	Unit	Range	Mean	Std	Missing rate
AQI	1	[3,500]	112.34	86.15	10.67%
PM _{2.5}	$\mu\text{g} / \text{m}^3$	[1,666]	81.44	78.00	11.54%
PM ₁₀	$\mu\text{g} / \text{m}^3$	[1,1000]	112.27	89.48	34.02%
O ₃	$\mu\text{g} / \text{m}^3$	[1,356]	58.72	57.71	13.91%
SO ₂	$\mu\text{g} / \text{m}^3$	[1,297]	18.54	25.98	11.88%
NO ₂	$\mu\text{g} / \text{m}^3$	[1,270]	56.69	34.51	12.49%
CO	mg / m^3	[0.1,10]	1.26	1.14	12.60%

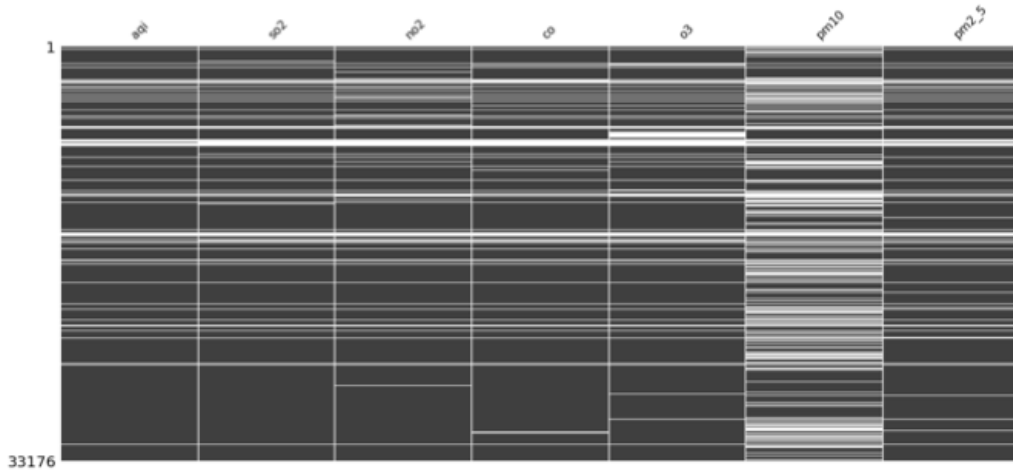


Figure 4.3 The visualization of the air quality data (Guanyuan)

Notably, the missing rate of PM₁₀ is much higher than other features, which reaches 34.02%. In order to distinguish whether the missing rates and missing intervals of PM₁₀ for other monitoring stations are similar, the PM₁₀ data at all sites were visualized in Figure 4.4.

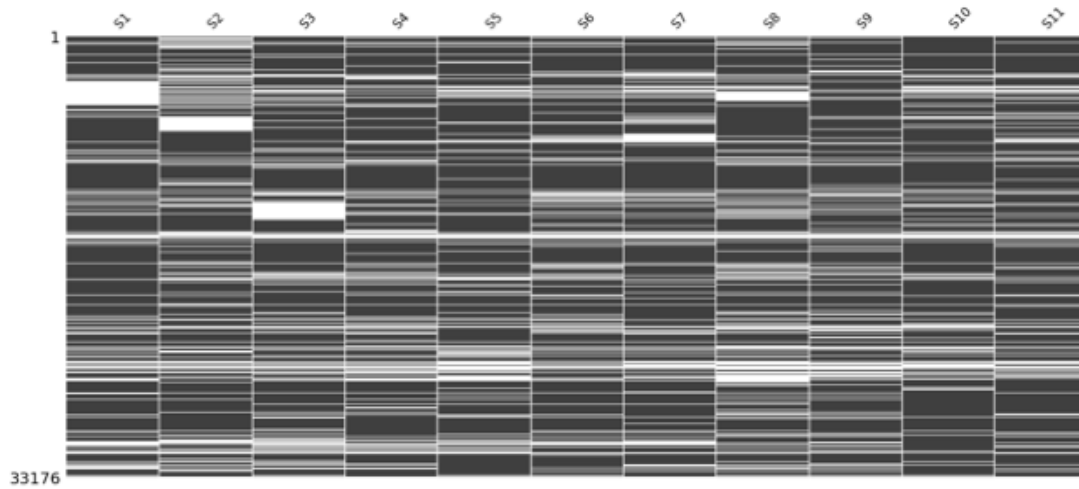


Figure 4.4 The visualization of the PM10 Data at stations

From Figure 4.4, it is obvious that the missing parts appeared at different stations are distinct at most of the time. According to Table 4.4, the missing rates of PM10 at all stations are higher than 30%. We will discuss how to complete such a series in section 4.5.

Table 4.4 The amount of the PM10 data at stations

Station	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Available Data	21057	19534	20406	21112	21476	21892	20931	19508	20556	22523	22369
Missing Rate (%)	36,53	41,12	38,49	36,36	35,27	34,01	36,91	41,20	38,04	32,11	32,57

4.3.3 Spatiotemporal Correlation Analysis

The spatial correlation of the PM2.5 data among the selected stations was measured by Pearson's correlation coefficient and the results are shown in Table 4.5. All correlation values are above 0.79, which demonstrate that strong spatial correlations exist among these selected stations.

Table 4.5 The Person's coefficients between stations

R	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
S1	1.0	0.79	0.95	0.97	0.94	0.94	0.9	0.88	0.82	0.81	0.89
S2	0.79	1.0	0.81	0.8	0.81	0.83	0.84	0.83	0.87	0.91	0.84
S3	0.95	0.81	1.0	0.96	0.96	0.96	0.91	0.89	0.84	0.83	0.9
S4	0.97	0.8	0.96	1.0	0.96	0.95	0.91	0.89	0.83	0.82	0.89
S5	0.94	0.81	0.96	0.96	1.0	0.94	0.9	0.89	0.83	0.83	0.89
S6	0.94	0.83	0.96	0.95	0.94	1.0	0.95	0.88	0.85	0.84	0.93
S7	0.9	0.84	0.91	0.91	0.9	0.95	1.0	0.87	0.85	0.86	0.92
S8	0.88	0.83	0.89	0.89	0.89	0.88	0.87	1.0	0.9	0.82	0.86
S9	0.82	0.87	0.84	0.83	0.83	0.85	0.85	0.9	1.0	0.86	0.86
S10	0.81	0.91	0.83	0.82	0.83	0.84	0.86	0.82	0.86	1.0	0.86
S11	0.89	0.84	0.9	0.89	0.89	0.93	0.92	0.86	0.86	0.86	1.0

Then, the autocorrelation function was used to evaluate the temporal correlation of PM2.5 data at each station. For time delay d , the formulation of the autocorrelation coefficient is as following:

$$\rho_d = \frac{Cov(y(t), y(t+d))}{\sigma_{y(t)} \sigma_{y(t+d)}}$$

where $y(t)$ and $y(t+d)$ indicate the PM2.5 data at time t and time $t+d$ respectively, $Cov(\cdot)$ represents the covariance and σ denotes the standard deviation. The autocorrelation coefficients of each station are shown in Figure 4.5, in which the 11 curves correspond to different stations respectively. According to the figure, with the increase of time lag d , it has less influence on the status at time t . What's more, when the time lag d is smaller than 18 ($d < 18$), the autocorrelation coefficient is above 0.5, which indicates a strong correlation in time. These findings can be considered as reference for determining input intervals. How to determine the most appropriate input period will be discussed in section 5.1.3.

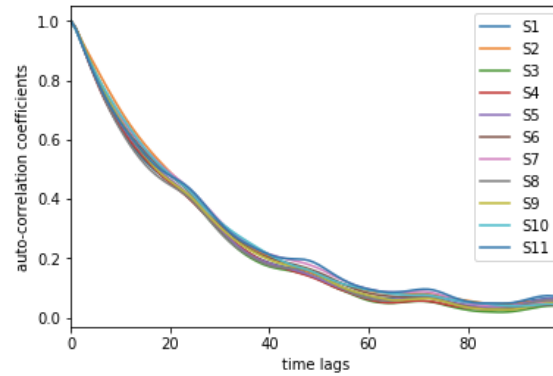


Figure 4.5 The autocorrelation coefficients of each station

4.4 Dataset for Visibility Predictions

Since the missing rate of visibility in the above PM2.5 dataset is pretty high, which reaches 44.86%, this dataset cannot be used for visibility forecasts anymore. Instead, another dataset was selected for visibility predictions. Compared with the PM2.5 dataset, which lasts for almost four years, the newly introduced dataset is about half of it. The hourly meteorological and air quality data from 2016/01/01 00:00 to 2017/12/31 23:00 in station 54511 was obtained. This dataset was from the historical data archive of the China Meteorological Agency (CMA). ‘54511’ is the WMO id of this station and it refers to the Beijing Capital International Airport.

Figure 4.6 gives the data visualization results. Similarly, white parts indicate missing parts. According to Figure 4.6, the missing rates of all features are pretty low. Hence, this dataset was completed by linear interpolation immediately. The statistical summary of all features is given in Table 4.6.

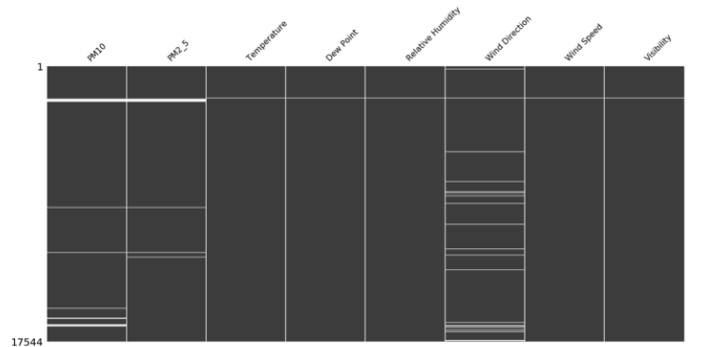


Figure 4.6 The visualization of the visibility dataset

Table 4.6 The statistical summary of the visibility dataset

Variable	Unit	Range	Mean	Std	Missing rate
PM _{2.5}	$\mu\text{g} / \text{m}^3$	[1,1000]	83.17	78.39	3.24%
PM ₁₀	$\mu\text{g} / \text{m}^3$	[25,1000]	106.86	90.02	1.89%
Temperature	$^{\circ}\text{C}$	[-16,42]	14.06	11.64	0.24%
Dew point	$^{\circ}\text{C}$	[-36,27.7]	2.21	14.11	0.24%
Relative Humidity	%	[5,99]	50.63	24.70	0.24%
Wind direction	($^{\circ}$)	[0,360]	166.02	103.28	3.71%
Wind speed	m/s	[0,9.5]	2.09	1.36	0.24%
Visibility	m	[23,35000]	13763.56	11668.76	0.26%

4.5 Data Preprocessing

Air quality features with small missing rates were completed by the linear interpolation. However, for the features with high missing rates (above 20%), fixing them in this simple way is likely to add large bias to the dataset itself. Because the missing rates of PM10 at all stations are higher than 30%, their missing parts should be fixed in a different way. Considering the strong spatial correlations among stations and the obvious autocorrelation in each site, k-Nearest Neighbor (kNN) [43] imputations were used when the missing interval is large and linear interpolations were implemented when the missing granularity is small. To summarize, the missing values were fixed through three steps.

Step1: Linear interpolations while the missing period is smaller than 3 hour.

Step2: kNN imputations when the number of available data at nearby stations is larger than 3.

Step3: Linear interpolations for the series after 1) and 2).

kNN imputations select stations that are close to the station of interest to fix missing values. If the value in timestamp t at station A (x_A) is missed, this method would find K other stations, which have a value x_i presented in timestamp t with a

small distance to station A . An inverse distance weighted average of values at timestamp t from these K geo-graphically closest stations is then used as an estimate for x_A . The specific formula is given as follows.

$$x_A = \sum_{i=1}^K \lambda_i x_i$$

$$\lambda_i = \frac{1}{d^2(x, x_i)} / \sum_{i=1}^K \frac{1}{d^2(x, x_i)}$$

The distance here refers to the straight-line distance between two sites.

After applying the method mentioned above to fix the missing values in the PM10 series for all stations, Table 4.7 shows how the number of valid data instances changes after each step. The missing parts are completed gradually, which demonstrates the effectiveness of each step.

Table 4.7 The number of valid PM10 data instances after each step

	Available data at all stations	Missing data at all stations
Origin	6730	3762
After Step1	12644	1685
After Step2	29811	1685
After Step3	33176	0

4.6 Summary

In this chapter, considering the accessibility of data, we introduced the methodologies for PM2.5 and visibility predictions using deep learning separately. In section 4.1, the PM2.5 predictions problem was formulated. For the short-term forecast (12 hours in advance), data of all selected features of the target station along with PM2.5 data of three surrounding stations were injected. With this framework, spatial and temporal correlations are considered by neighboring stations and the ‘memory’ of LSTM respectively.

While air quality data were obtained from 11 air quality monitoring stations, meteorological data was accessed at just one point. The problem description of

visibility predictions was shown in section 4.2. In this task, data of all selected features of the target station were considered. The network output is the predicted visibility level 4 hours ahead.

Afterward, the data visualization and statistical summary of these two datasets were finished separately in section 4.3 and 4.4. Since air quality data was obtained from multiple sites, spatiotemporal correlation analysis among these monitoring stations was implemented in section 4.3.3. Besides, for air quality variables with high missing rates, a data imputation method that fully considers temporal and spatial correlations was put forward in section 4.5.

5. Performance Evaluation

In chapter 3, the theoretical knowledge of deep learning that will be mentioned in this chapter was provided. In addition, the methodologies for PM2.5 and visibility predictions using deep learning were introduced in chapter 4. In this chapter, we will configure and tune the LSTM frameworks to our prediction problems and then evaluate their performance for three different tasks: PM2.5 predictions, PM10 predictions and visibility predictions in section 5.1, 5.2 and 5.3 separately.

5.1 PM2.5 Predictions at station Guanyuan, Beijing

In this section, the spatiotemporal LSTM model for PM2.5 predictions in station Guanyuan, Beijing will be built. For a neural network, many hyperparameters are required to be configured, including those that specify the structure of the network itself and those that determine how the network is trained. These two parts correspond to gradient descent hyperparameters and model hyperparameters. Besides, feature engineering will be implemented to select appropriate features. After network configuration, the proposed spatiotemporal LSTM model will be compared with Lotos-Euros and another spatiotemporal prediction framework.

5.1.1 Mini-Batch Gradient Descent Hyperparameters

When training a neural network, results will depend not only on the chosen network structure but also on the selected training method. The training method itself can have many hyperparameters. As described in section 3.4.1, mini-batch gradient descent was selected and the hyperparameters involved include learning rate, loss function, mini-batch size, number of training iterations and so on. These fixed training details are described in Table 5.1.

Table 5.1 The mini-batch gradient descent hyperparameters

Parameter	Value
Number of records	33176
Training set	69.34%
Validation set	17.34%
Test set	13.32%
Parameter update	Adam
Batch size	128
Loss function	Mean Squared Error
Activation function (Output layer)	None

After data preprocessing, hourly data from 2013/01/18 16:00 to 2016/10/31 23:00 was obtained completely. Totally, there are 33176 records. For the records from 2013/01/18 16:00 to 2015/04/30 23:00, we took the first 80% as training set and the last 20% as validations set. The data from 2016/05/01 00:00 to 2016/10/31 23:00 was considered as the test set. Since the PM2.5 prediction is a regression task, the target is to minimize the average errors between the observed concentrations and the predicted ones, which is exactly what Mean Squared Error (MSE) represents.

When it comes to the choice of the batch size, with an appropriate batch size, updates can be computed more efficiently due to the use of parallel architectures. However, the batch size cannot be too large. Keskar N S et al. (2016)[44] pointed out that when using a larger batch there is a degradation in the quality of the network, in terms of its ability to generalize. The lack of generalization ability is due to the fact that *large-batch methods tend to converge to sharp minimizers of the training function while small-batch methods converge to flat minimizers*. Herewith, considering the size of the training set, which is 23004, batch size was set to be 128. That is, the parameters update 180 times in each epoch. Besides, Adam method was chosen to adjust learning rates when updating weights. The related parameters in Adam follow those provided in the original paper[41], which are also given in section 3.4.2. We do not use any activation function in the output layer of this regression model because any activation function is likely to add a transformation to

the numerical outputs and put restrictions on them.

5.1.2 Feature Selections

Before doing experiments to select the best model parameters, we will determine the input features in this section. According to section 4.1, spatial correlations are considered by adding neighboring PM2.5 concentrations. In the case study of station Guanyuan, data at S1, S3, and S7 were considered. The effectiveness of containing nearby stations will be demonstrated by comparing it with the identical configured network but discarding nearby PM2.5 data as input.

Another concern is about visibility. As described in section 4.3.1, the missing rate of visibility is very high, at 44.86%. Since we just got accessibility to meteorological data at one location, visibility cannot be completed by kNN imputations like PM10. In our research, it was completed by linear interpolation immediately. However, in the meanwhile, according to the literature study on visibility in section 2.5, there is a strong relationship between PM2.5 and visibility. In order to distinguish whether it is necessary to contain visibility as an input feature, a similar method as evaluating adding PM2.5 data at nearby stations was used.

Because the mini-batch gradient descent hyperparameters were already determined in section 5.1.1, in order to make the above comparisons, the model hyperparameters are also required. *In this section, the model hyperparameters were set for comparative reasons and the best model hyperparameters will be discussed in the next section.* The specific structure of this network is shown in Table 5.2.

In our experiments, whether the number of epochs is suitable for converging were checked through comparing the training loss and validation loss in each epoch. Through experiments, the appropriate number of iteration times is 50. As described in section 3.3, LSTM neural networks with a large number of parameters easily face the problem of overfitting and dropout is a powerful method to solve it.

Table 5.2 Structure of the configured LSTM framework

Parameter	Value
Input interval	12 hours
Number of hidden layers	2 LSTM layer
Neurons per layer	100
Epoch	50
Dropout rate in input	1 st layer: 0; 2 nd layer: 0
Dropout rate in recurrent connections	1 st layer: 0; 2 nd layer: 20%
Dropout rate in output	1 st layer: 0; 2 nd layer: 20%

After setting the model hyperparameters, the related results in the validation set are given in Table 5.3. The second and third lines of Table 5.3 give the comparison of adding nearby PM2.5 concentrations while the third and fourth lines show the comparison of containing visibility. According to the performance indicators RMSE and MAE, within the identically configured network, considering PM2.5 at nearby stations and visibility as input significantly increase prediction performance. Therefore, in the following work, input features are fixed to be 15.

Table 5.3 Performance measures used for feature selections

Input Features	RMSE	MAE
11	38.97	25.80
14(add PM2.5 at three nearby stations)	36.82	23.91
15(add visibility)	33.18	23.02

5.1.3 Model Hyperparameters

The input features have already been determined by feature engineering. In this section, best model hyperparameters including input intervals, number of hidden layers and number of hidden units in each hidden layer will be determined by hyperparameter space exploration (see Table 5.4). Two common exploration methods are coordinate descent and grid searches.

Table 5.4 Experimental parameters for the spatiotemporal LSTM framework

Parameters	Value Set
Number of layers	2,3
Number of nodes (per layer)	50,100,200
Input time intervals	12,18,24,30,36,42,48

In order to find the best values for the number of hidden layers and nodes per hidden layer, grid searches were used. Grid searches try every hyperparameter setting over a specified range of values and therefore involve a cross-product of all intervals. The corresponding network performances (RMSE) in the validation set are shown in Table 5.5. The minimal RMSE is 33.01 and it appears in which the number of layers is 2 and the number of nodes per layer is 150. However, compared with the RMSE when the number of layers is 2 and the number of nodes per layer is 100, which is 33.18, results do not improve significantly.

Table 5.5 Grid searches for the best model hyperparameters

Number of Nodes	Number of Layers	
	2	3
50	35.72	34.12
100	33.18	33.78
150	33.01	34.23

In the meanwhile, according to Table 5.6, when the number of layers is 2, adding 50 LSTM neurons per layer will add 153,450 parameters in total. Considering the training efficiency, we fixed the number of layers to be 2 and the number of neurons per layer to be 100, which are identical to the network structure given in Table 5.2. Within this structure, the number of parameters is 126,901 in total.

Table 5.6 The number of parameters for two different structures

Number of Layers	Number of Nodes(per layer)	Number of Parameters
2	100	126,901
2	150	280,351

For the hyperparameter input intervals, the coordinate descent was used, which means that we kept all hyperparameters fixed except for input intervals, and adjusted

it to minimize the validation error. The corresponding RMSE and MAE are shown in Table 5.7, in which the RMSE stays about the same (within 2). This reveals that the granularity of the short-term data is significant enough to capture changes in air pollution. Similar findings were mentioned by Reddy V (2018) [2], in which the LSTM sequence-to-sequence model was used for PM2.5 predictions. Reddy V tried to vary past time steps from 20 to 60 to predict further 5 hours and the resulting RMSE changes within 5. So for our prediction task, the most appropriate input interval is 18.

Table 5.7 Coordinate descent for the best input length

Input intervals	12	18	24	30	36	42	48
RMSE	33.18	31.61	32.26	32.42	32.45	32.56	31.91
MAE	23.02	21.97	22.28	22.69	22.78	22.33	22.41

To summarize, the final spatiotemporal LSTM model architecture is given in Figure 5.1.

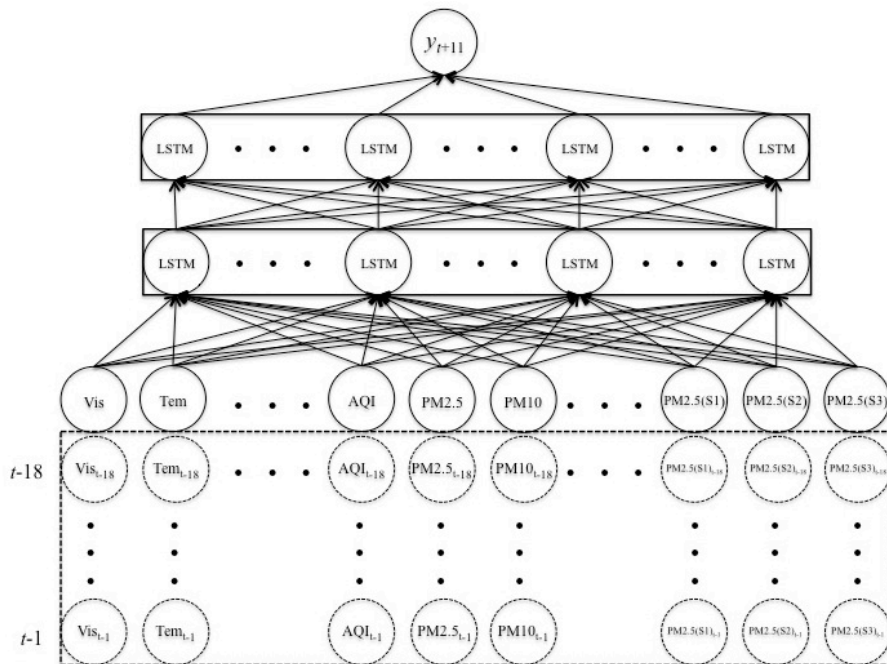


Figure 5.1 The architecture of the configured LSTM framework

5.1.4 LSTM against Lotos-Euros

We assessed the performance of the proposed spatiotemporal LSTM model against Lotos-Euros, which is based on CTMs. As mentioned in section 2.2.1, Lotos-Euros is a regional model over Europe and is designed on a regular longitude-latitude grid ($0.5^\circ \times 0.25^\circ$). However, it can be applied anywhere and with arbitrary grid resolution in terms the horizontal resolution is larger than about 3 km[15].

The reason why the horizontal resolution should be larger than 3 km lies in the vertical design of Lotos-Euros. In the vertical direction, the model consists of a static surface layer of 25 m, a dynamic layer extending from 25 m to the top of the mixing layer, and three dynamic reservoir layers that all together fills the vertical between the top of the mixing layer to 5 km altitude. This specially designed vertical structure helps a lot in solving the chemical process that is most time consumed in a very efficient way. Nevertheless, if higher resolution is desired, the horizontal and vertical dimension could be out of balance for the used parameterizations and more layers have to be added within the mixing layer[14].

Lotos-Euros as a CTMs-based model, predicts PM2.5 in sequence (usually 24 hours). Take a specific day for example, Lotos-Euros exported the PM2.5 concentrations on 2016/10/03 from 00:00 to 23:00 at a time on 2016/10/02 at 23:00. As for our spatiotemporal LSTM model, PM2.5 concentrations are predicted 12 hours in advance. We outputted the PM2.5 concentrations on 2016/10/03 at 00:00 on 2016/10/02 at 12:00 and concentrations at 01:00 on 2016/10/02 at 13:00 and so on. Table 5.8 shows this comparison intuitively.

Since Lotos-Euros exports 24-hour PM2.5 concentrations at a time while the spatiotemporal LSTM model predicts PM2.5 concentrations 12 hours in advance, the average prediction length of these two methods is equal.

Table 5.8 Running timestamp for two methods

Target timestamp	Running timestamp of Lotos-Euros	Running timestamp of the proposed LSTM framework
2016/10/03 00:00	2016/10/02 23:00	2016/10/02 12:00
2016/10/03 01:00	2016/10/02 23:00	2016/10/02 13:00
2016/10/03 02:00	2016/10/02 23:00	2016/10/02 14:00
...
2016/10/03 22:00	2016/10/02 23:00	2016/10/03 10:00
2016/10/03 23:00	2016/10/02 23:00	2016/10/03 11:00

As for the location used for comparison, the predicted PM2.5 concentrations using Lotos-Euros in the grid point (116.375, 39.9375) were extracted. As illustrated above, the horizontal resolution of Lotos-Euros cannot be smaller than 3 km and the selected grid point is just 1.31km far away from the Guanyuan station (see Table 5.9), we can ignore this distance and consider them as the same point.

Table 5.9 The location of Guanyuan station and Lotos-Euros grid point

Guanyuan monitoring station	Lotos-Euros grid point	Distance
(116.361 , 39.9425)	(116.375 , 39.9375)	1.3182km

To summarize, *the average prediction length of these two methods are both 12 hours and the selected Lotos-Euros grid point can be considered as in the same location as station Guanyuan.* Therefore, we can show the powerful performance of our spatiotemporal LSTM model through comparing it with Lotos-Euros. The prediction performance of these two methods in the test set (2016/05/01 00:00 - 2016/10/31 23:00), i.e. half-year period are shown in Figure 5.2 and Figure 5.3.

Intuitively, in Figure 5.2, *a point locates in the line $y=x$ means that its observed value y equals to its predicted value x , which can be considered as an ideal prediction.* A point far away from this straight line implies that its predicted value far away from its recorded value. *A point located in the zone above the line $y=x$ ($y>x$) corresponds to underestimation while a point located in the zone under the line $y=x$ ($y<x$) means overestimation.*

R. Timmermans et al. (2017) [16] pointed out that there is a systematic

underestimation of particulate matter concentrations in Beijing using Lotos-Euros. This can be reflected in the left scatter plot of Figure 5.2 because many data points appear in the zone $y > x$. Besides, many data points are located in the area $y > x$ and $y > 200$. This indicates the underestimate phenomenon of Lotos-Euros, especially in serious air pollution situations. Compared with the scatter plot of Lotos-Euros, data points are more concentrated on both side of $y = x$ in the right scatter plot and fewer data points appear in the area $y > x$ and $y > 200$, which shows the competitiveness of the proposed spatiotemporal LSTM model in PM2.5 predictions.

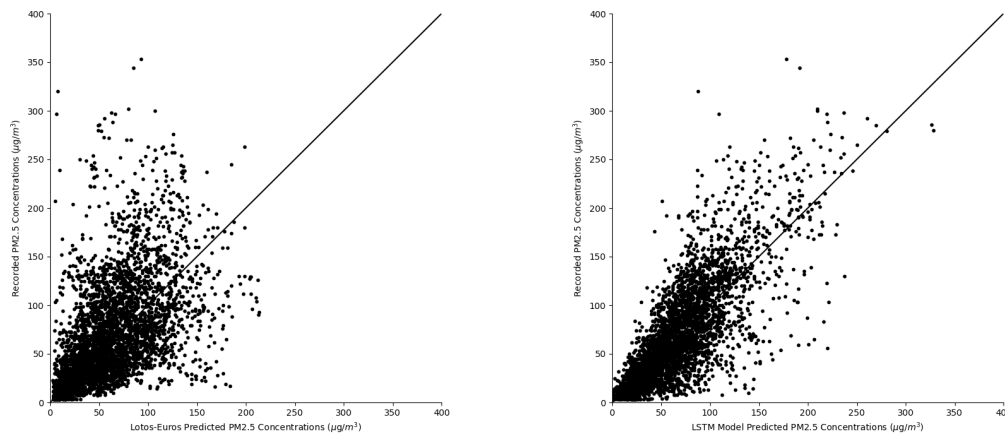


Figure 5.2 (left) The scatter plot of Lotos-Euros in PM2.5 predictions; (right) The scatter plot of the proposed LSTM framework in PM2.5 predictions

As mentioned in section 3.5.1, the regression results can be classified into their corresponding air quality levels and therefore confusion matrix can be utilized. While Figure 5.2 compares the predictions through scatter plotting, Figure 5.3 shows this comparison from quantitative perspectives through confusion matrices.

For a confusion matrix without normalization, values in the main diagonal correspond to the total times when the recorded air quality levels are identical to the predicted air quality levels, which indicate the ideal prediction times. The values under the main diagonal show how the predictions are underestimated corresponding to their measured levels while the values above the main diagonal imply overestimation. *Therefore, an ideal confusion matrix is a diagonal matrix, in which the entries outside the main diagonal are all zero. The non-zero entries farther away from the main diagonal imply the weaker prediction performance.*

As for a confusion matrix with normalization, each value in the matrix is divided by the sum of the located row and the sum of a row is the number of records in this level. *Therefore, values in the main diagonal correspond to recall rates for various levels.*

According to Figure 5.3 a), which shows the confusion matrix of Lotos-Euros, since L3, the values in the main diagonal are much smaller than the ones under the diagonal. This reflects the underestimation phenomenon of Lotos-Euros, especially in terms of high air quality levels, which is consistent with what we discovered in the left scatter plot of Figure 5.2. Take L4 for example, the entry in the diagonal is 52, however, the entries under the diagonal in this row are 150, 136 and 34 respectively. These indicate that L4 records were predicted as L3 for 150 times, as L2 for 136 times and L1 for 34 times.

When it comes to Figure 5.3 b), which corresponds to the confusion matrix of the spatiotemporal LSTM model, values in the main diagonal are larger than those in Figure 5.3 a) from L2. Because two confusion matrices are based on the same period, the numbers of instances at different levels for these two matrices are identical. *Larger entries in the main diagonal show better performance.*

What's more, the comparison is made considering the most severe air quality level (L6), which is described as hazardous. In the selected half-year period, 33 instances were recorded as L6. Lotos-Euros predicted it as L2 for 12 times (most often) and L3 for 11 times. But L2 is considered as moderate while L3 is described as unhealthy for sensitive groups, which again shows the weak performance of Lotos-Euros in predicting extreme situations. As for our proposed LSTM model, even though the value in the diagonal at L6 is small, it was predicted as L5 for 23 times and L5 is thought to be very unhealthy. These discussions reveal that in addition to comparing the diagonal elements, the off-diagonal values closer to the main diagonal indicate better but not ideal predictions while farther away from the main diagonal corresponds to worse prediction performance.

Similar findings can be obtained according to the confusion matrix with normalization. As for the normalized one corresponding to Lotos-Euros (Figure 5.3 c)), the recall rates at L4 and L5 are pretty low, at 13% and 7% respectively and the

recall rate at L6 reaches 0. Besides, the percentages in the zone where the recorded level is over L3 while the predicted level is below L3 i.e. under the main diagonal are rather high. All these observations reflect the problem of systemic underestimation using Lotos-Euros. The normalized confusion matrix of the proposed LSTM model is given in Figure 5.3 d). Without a significant difference in the recall rate of L1, starting from L2, the recall rates of all levels are higher than Lotos-Euros, indicating that the proposed framework can predict the levels accurately much more often.

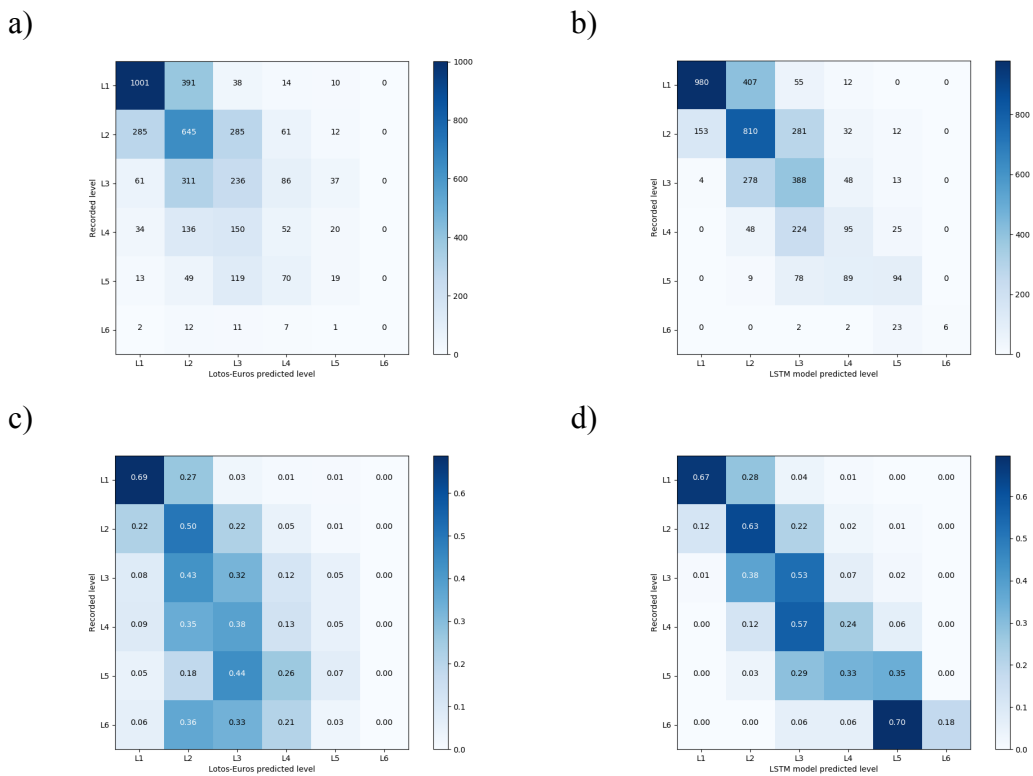
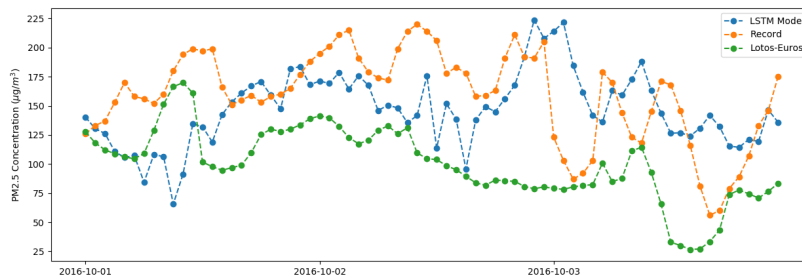


Figure 5.3 a) The confusion matrix of Lotos-Euros; b) The confusion matrix of the proposed LSTM framework; c) The confusion matrix of Lotos-Euros with normalization; d) The confusion matrix of the proposed LSTM model with normalization.

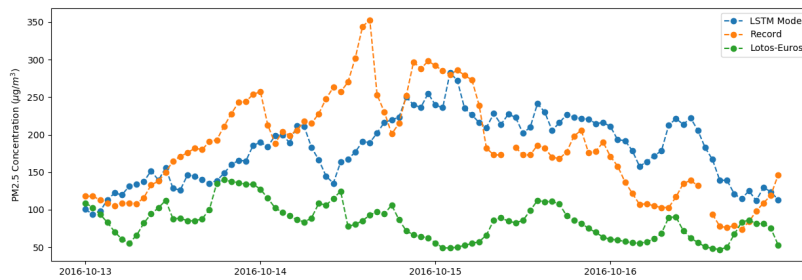
According to Figure 5.2 and 5.3 and the above analysis, the proposed spatiotemporal LSTM model can overcome the systematic underestimation problem that Lotos-Euros encounters to some extent. In addition to scatter plots and confusion matrices, hourly plotting at a shorter period is desired so as to show the predictions intuitively. Since much more attention is paid to severe situations and many ‘very unhealthy’ records ranging from 150 to 250 (L5) appeared in 2016/10. We plotted

the results in three successive periods, in which the ‘very unhealthy’ record existed in each day in Figure 5.4. Afterward, the hourly plotting at a longer period (half month) is depicted in Figure 5.5.

a)



b)



c)

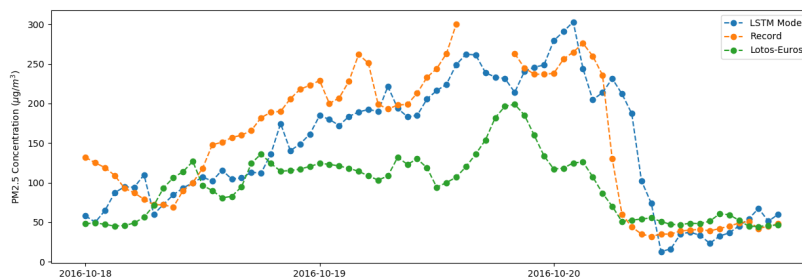


Figure 5.4 The PM2.5 concentrations in three different periods: a) 2016/10/01-2016/10/03; b) 2016/10/13-2016/10/16; c) 2016/10/18-2016/10/20.



Figure 5.5 The PM_{2.5} concentrations in four different time period: a) 2016/06/01-2016/06/15, b) 2016/07/01-2016/07/15,c) 2016/08/01-2016/08/15,d) 2016/09/01-2016/09/15

Following the first picture (Figure 5.4 a)), both the proposed framework and Lotos-Eurois do not perform well. The reason may lie in the fact that the PM_{2.5} concentrations fluctuated a lot in this three-day period, which makes the task much more difficult. When it comes to the second and third picture (Figure 5.4 b) and c)),

considering the long prediction intervals (12 hours) and the complicated underlying interactions of PM2.5, the plotting results of the spatiotemporal framework are acceptable because at most of the time, it can capture the trends and the gaps between the predictions and the measures are not very large. As for the plotting of Lotos-Euros, the underestimation phenomenon is very obvious since many measures from 150 to 250 (L5) were predicted to the range between 75 and 150 (L3 and L4). This is consistent with the confusion matrix shown in Figure 5.2 c). In that confusion matrix, 44 percentages of L5 records were predicted as L3 while 26 percentages of records were predicted as L4.

Additionally, the hourly plotting at half-month periods is shown in Figure 5.5. According to these figures, the LSTM can achieve acceptable predictions at most of the time. However, for the sharp reductions in the real world, the LSTM plotting shows a slight ‘time delay’ phenomenon. Similar degradation phenomenon was found by Reddy V (2018) [2], in which the LSTM sequence-to-sequence model was employed for PM2.5 predictions, with 30 time steps previous to 10 future time steps.

5.1.5 Comparison with Other Spatiotemporal Prediction Framework

In addition to comparing the proposed spatiotemporal LSTM model with a CTMs-based model like Lotos-Euros, we also compared it with a deep learning based model. Fan J et al. (2017) [9] proposed a similar LSTM-based prediction framework for Jing-Jin-Ji area, China, which also considers spatial correlations among stations and aims at predicting PM2.5 8 hours in advance. The experiments showed that their proposed framework outperformed both deep feedforward neural networks (DFNN) and gradient boosting decision trees (GBDT). Table 5.10 shows the comparisons between our proposed framework and Fan’s framework in terms of network hyperparameters and performance indicators.

Table 5.10 Framework Comparisons

	The proposed LSTM framework	Fan's prediction framework ^[9]
Input Length	18	48
Prediction Length	12	8
Features	15	19
Layer	2 LSTM layers	2 fully connected layers & 2 LSTM layers
Dataset Period	2013/01-2016/10	2013/09-2015/01
Data Preprocessing	kNN and linear imputations	forward-fix
RMSE	31.72	35.74
MAE	22.01	23.72

According to the experiments in section 5.1.3, increasing the input size from 18 to 48 do not increase the performance noticeably and *a short-term input interval is enough for capturing changes in air pollution*. Besides, in addition to air quality properties and meteorological properties, time properties e.g. weekday, date, month and hour and spatial properties, e.g. longitude and latitude of stations were considered as input features in Fan's framework. However, the RMSE and MAE revealed that adding these features do not help a lot and therefore *air quality and meteorological features are enough for air pollutant forecast*.

Another important difference between these two frameworks is the dataset size and the data preprocessing method. Since the air quality monitoring stations at Jing-Jin-Ji area were built in 2013, all accessible air quality data at monitoring stations are available from 2013. What's more, Fan fixed the missing values using the latest valid observation while kNN imputations were used during data preprocessing in our framework. With larger training dataset and kNN imputations, even though our framework structure is simpler than Fan's framework and the prediction task is longer, the RMSE and MAE are smaller.

5.2 Transfer Learning for PM10 predictions

Transfer learning, as the name states, requires the ability to transfer knowledge from one domain (source) to another (target). It has been used in a large variety of domains including Natural Language Processing (NLP) and Computer Vision[27]. In our supervised learning task, we consider transferring knowledge from PM2.5 predictions to PM10 predictions. Since both source data and target data are labeled, transfer learning is realized through model fine-tuning.

Here, the input features of PM10 predictions are the same as the ones used for PM2.5 predictions but the target is changed into PM10 concentrations. Since the new dataset for PM10 predictions is large and similar to the original one, we fine-tuned through the whole network.

The framework obtained in Section 5.1.3 was considered as a pre-trained network and model fine-tuning was done by continuing the mini-batch gradient descent. One strong advantage of this method is that it can accelerate the training process and the network does not need to learn from the randomly initialized weights. The experimental results shown in Table 5.11 reveal how can fine-tuning method increase the training efficiency. If network fine-tuning is applied, in order to obtain similar results, the number of iterations required is 5 rather than 50.

Table 5.11 Performance Evaluations

	The LSTM model without fine-tuning	The LSTM model with fine-tuning
Epoch	50	5
RMSE	48.91	49.12
MAE	30.93	31.62

The scatter plot of the fine-tuned framework in the test set, which is in the same period as the one used in PM2.5 predictions (2016/05/01 00:00 - 2016/10/31 23:00) is shown in right picture of Figure 5.6. Again, the left picture corresponds to the scatter plot of Lotos-Euros.

a)

b)

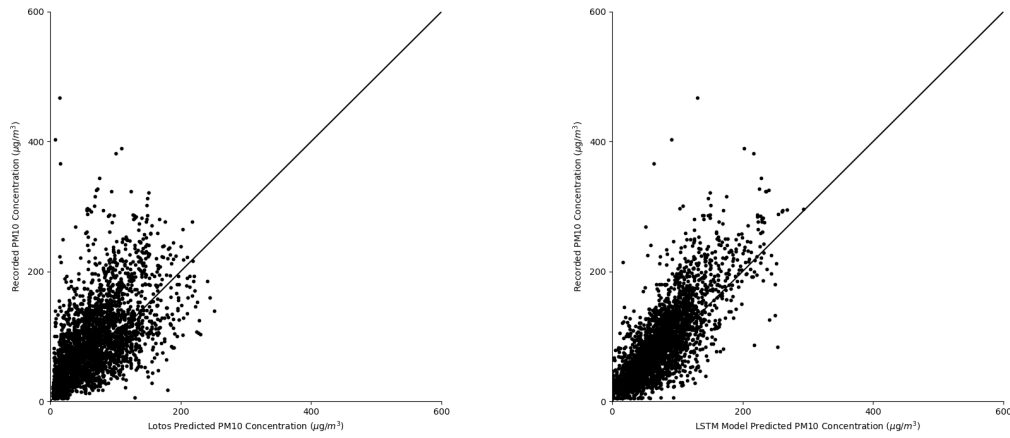


Figure 5.6 (*left*) The scatter plot of Lotos-Euros in PM10 predictions; (*right*) The scatter plot of the fine-tuned framework in PM10 predictions.

It is noteworthy that an outlier exists in this selected period, reaches $1000 \mu\text{g}/\text{m}^3$. This outlier was deleted in these scatter plots so as to restrict the value range in both x-axis and y-axis. The scatter plots of PM2.5 predictions were already depicted in section 5.1.4. Similarly, a point locates in the line $y = x$ corresponding to an ideal prediction and a point farther away from this line indicates a worse prediction. Compared with the left picture of Figure 5.6, data points are more concentrated on both side of $y = x$ in the right figure.

In conclusion, the proposed spatiotemporal LSTM model for PM2.5 predictions can be easily transferred to PM10 predictions. Besides, compared with Lotos-Euros, the LSTM framework not only have competitive performance in PM2.5 predictions but also in PM 10 predictions.

5.3 Visibility Predictions at Beijing Capital International

Airport

As illustrated in section 4.4, the value of visibility ranges from 23 to 35000 and its standard deviation reaches 11668, which indeed indicate the extremely high variability property of visibility data. Besides, according to section 5.3.1, the visibility predictions face the problem of class-imbalance. Since the objective of this

research is to forecast visibility at airports, that is, to predict visibility levels for different kinds of aircraft to take off so that airport operators are able to take measures to reduce the economic loss and passengers inconvenience caused by the air traffic disruption. Considering the high variability property of visibility data and the research objective, the visibility prediction is considered as a classification task.

5.3.1 Dataset Partition

As described in section 4.4, there are 17544 available records (2016/01/01 00:00 to 2017/12/31 23:00) in the visibility dataset. Looking at Table 5.11, the total number of records in L1 and L2 are very low. The percentages of instances at L1 and L2 are 2.33% and 6.03% respectively. According to the definition, the percentage of an event that is smaller than 5% can be described as a *rare event*.

We split this dataset into three parts, aiming at keeping the proportions of different levels at these three sets to be consistent. With this partition, the LSTM model can learn the patterns from the training set; validation set can be used to determine the hyperparameters while test set for performance evaluation. Even though we kept the data distribution in these three sets to be similar, visibility forecasts still face the problem of *class-imbalance* because the percentages of instances at L1 and L2 are small.

Table 5.11 Visibility Dataset Partition

	L1	L2	L3	Sum
Train set	312	792	10530	11634
Validation Set	41	101	2815	2957
Test Set	56	165	2717	2938
Total	409	1058	16062	17529

5.3.2 Framework Parameters

Most of the mini-batch gradient descent hyperparameters and model hyperparameters in visibility task are identical as those mentioned in section 5.1.1 and 5.1.3. The hyperparameters changed are described in Table 5.12.

Table 5.12 Different Hyperparameters for Two Frameworks

	Visibility Predictions	PM2.5 Predictions
Prediction Length	4	12
Batch Size	64	128
Neurons (per hidden layer)	50	100
Neurons (Output Layer)	3	1
Activation Function (Output Layer)	Softmax	None
Loss Function	Weighted Cross-Entropy	Mean Square Error

Predicting airport visibility accurately four hours in advance can be extremely useful for passengers because they can have enough time to respond to airport delays and cancellations. Additionally, since the size of the visibility dataset is around half of the PM2.5 dataset, the number of neurons per layer was reduced to 50 and the batch size was changed to 64. Because the visibility forecast is a classification task and visibility values were divided into three classes, the number of neurons in the output layer is three and the softmax function was used. Since *class-imbalance* phenomenon was explored in the visibility dataset, the target function was weighted cross-entropy. One strong advantage of introducing weights is that it can add bias to events with small occurrence rates. The final LSTM model architecture for visibility forecast is given in Figure 5.6.

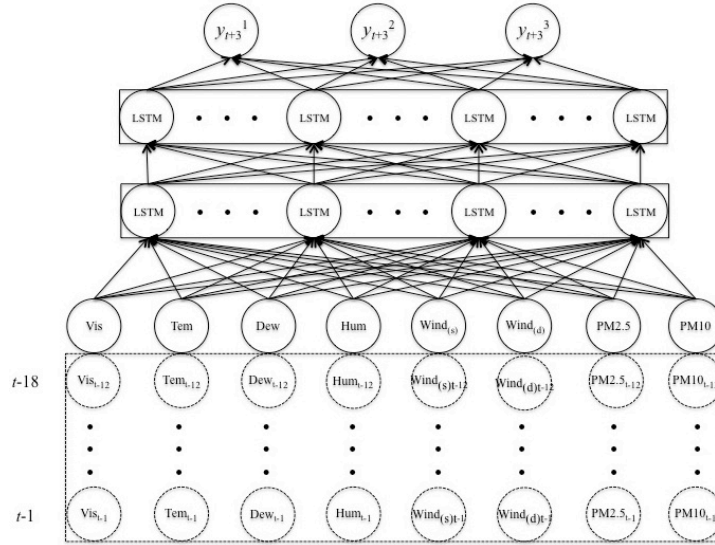


Figure 5.6 Architecture of the Configured LSTM Model for Visibility Predictions

After the above introduction, the undetermined hyperparameter is α_k . A coarse-to-fine sampling scheme will be employed to pick it. According to Table 5.11, $L1:L2:L3 \approx 1:2:40$. One intuitive attempt is to set the proportion of these three weights to be the inverse of the proportion of three levels i.e. $\alpha_1 : \alpha_2 : \alpha_3 = 40 : 2 : 1$, however, by experiments, this would reduce the recall rate of L3 seriously. Then, as a coarse sampling solution, we kept the weights $\alpha_1 : \alpha_2 = 2 : 1$ and set α_3 to be smaller.

By comparing the validation performance in terms of confusion matrices, $\alpha_1 : \alpha_2 : \alpha_3 = 20:10:1$ was the optimal setting in coarse sampling and after that, in fine sampling, the best proportion was $\alpha_1 : \alpha_2 : \alpha_3 = 18:10:1$. The corresponding confusion matrix in the test set will be discussed in the next section.

Table 5.13 The Coarse-to-fine sampling scheme

Coarse Sampling		Fine Sampling
10:5:1		16:10:1
20:10:1		18:10:1
30:15:1		20:10:1
		22:10:1

5.3.3 Prediction Results

In the last section, all hyperparameters involved have been discussed. The performance of the configured LSTM model in the test set is given in Figure 5.7.

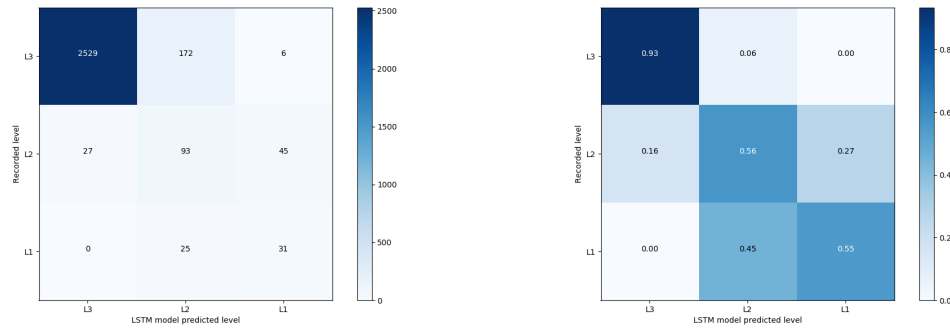


Figure 5.7 (*left*) The confusion matrix in visibility predictions; (*right*) The confusion matrix with normalization in visibility predictions

For visibility predictions, the total accuracy rate reaches 90.61%, which is calculated by dividing the sum of the values in the main diagonal by the sum of the values in the confusion matrix. According to the confusion matrix given in the left of Figure 5.7, the recall rate of L1 is 93% while its precision rate is 96%, indicating its superior prediction performance in the normal situations. When it comes to the low visibility level (L2 and L3) that can affect air traffic negatively, their recall rates are higher than 50%. That is, the numbers of correct predictions for each class (values in the main diagonal) are larger than the numbers of negative predictions for each class (non-diagonal entries). Considering the 4-hour prediction period and the extremely complicated mechanism of visibility, the above predictions are acceptable in practice.

5.4 Summary

In this chapter, three different prediction tasks including PM2.5 predictions, PM10 predictions, and visibility predictions were realized separately in section 5.1, 5.2 and 5.3. In the PM2.5 predictions, network hyperparameters were divided into gradient

descent hyperparameters, which determine how the network is trained and model hyperparameters that specify the structure of the network itself. Besides, in section 5.1.2, feature engineering was employed to select appropriate features. Best model hyperparameters were determined through experiments in section 5.1.3. After network configuration, the proposed spatiotemporal LSTM model was compared with Lotos-Euros in section 5.1.4 and another spatiotemporal prediction framework in section 5.1.5. All comparison demonstrated the competitive performance of the proposed framework.

In section 5.2, knowledge from PM2.5 predictions was transferred to PM10 predictions through fine-tuning. Through experiments, fine-tuning can improve the training efficiency significantly. Besides, it was shown that the LSTM framework also outperformed Lotos-Euros in terms of PM 10 predictions.

When it comes to visibility predictions, in section 5.3.1, the dataset was split into three parts (training set, validation set, and test set) and the proportions of different levels at these three sets are consistent. In addition, since ‘*class-imbalance*’ was explored, a coarse-to-fine sampling scheme was used to determine the best weights for the weighted cross-entropy in section 5.3.2. Finally, the performance of the configured LSTM model for visibility forecasts was given in section 5.3.3. Overall, the airport visibility predictions are acceptable because the total accuracy rate reaches 90.61% and the numbers of correct predictions for each level are larger than the numbers of negative predictions for each level.

6. Conclusions and Future Research

In this chapter, our studies on PM2.5, PM10 and visibility predictions are concluded. We look back at the research questions posed in Chapter 1, draw conclusions to the research objectives and give suggestions for future work.

6.1 Research Work Recap

In this research, methodologies to predict PM2.5, PM10, and visibility using LSTM NN were investigated. Different experiments were performed and several positive results were obtained. Our work consists of several main parts.

6.1.1 Dataset Analysis and Preprocessing

Dataset analysis and preprocessing is an important step in almost all big data and machine learning problems. Because missing data and confusing or incorrect data are common in large datasets, a proper analysis and preprocessing of the data is crucial for the success of a learning task. In the proposed methodology, for given data, the first step was to correct noise and errors and calculate the missing rate of each variable. Afterward, the dataset were visualized. With the help of visualizations, it was easier to evaluate the missing phenomenon of different features.

Since PM2.5 data were obtained from multiple stations, the spatial and temporal correlations were measured by Pearson's correlation coefficients and autocorrelation coefficients respectively. The strong spatial correlations among monitoring stations resulted in adding nearby PM2.5 data as input in the prediction framework while the autocorrelation coefficients were used as the reference for determining input intervals. Due to these strong spatiotemporal correlations, for air quality features with high missing rates, linear interpolations were implemented when the missing granularity is small and k-Nearest Neighbor (kNN) imputations were used when the missing interval is large.

6.1.2 Framework Configuration

For a neural network, network hyperparameters can be divided into gradient descent hyperparameters and model hyperparameters. In PM2.5 predictions, the gradient descent hyperparameters were investigated through the literature study. In addition, feature engineering was employed to select the appropriate features. As for the model hyperparameters, the best input interval was investigated by coordinate descent while grid searches were used to explore the most appropriate number of hidden layers and number of hidden units per hidden layer.

As for PM10 predictions, since transfer learning was applied through model fine-tuning, the only hyperparameter that needed to be decided was the number of iteration. We checked whether the number of epochs was suitable for converging through comparing the training loss and validation loss in each epoch. When it comes to the weights in the weighted cross-entropy used for visibility predictions, a coarse-to-fine sampling scheme was adopted.

6.1.3 Performance Evaluation

For PM2.5 predictions, the performance was compared with Lotos-Euros (a regional CTMs-based system) and an LSTM-based prediction framework (Fan J et al. (2017) [9]) that also considers spatial correlations among stations and aims at predicting PM2.5 8 hours in advance in a similar region respectively. The proposed framework was compared with Lotos-Euros through their scatter plots and confusion matrices in the half-year period. In addition, hourly forecasting results of these two methods were compared with records at three-day and half-month periods so as to show the prediction accuracy intuitively. RMSE and MAE were used when comparing with Fan's spatiotemporal prediction framework.

As for PM10 predictions, due to the application of transfer learning, the training efficiency was evaluated. After training, the LSTM framework was also compared

with Lotos-Euros according to their scatter plots. When it comes to visibility predictions, since it is a kind of classification problems, the forecast performance was evaluated through the confusion matrices.

6.2 Conclusions

To draw conclusions, the research objectives mentioned at the beginning of this thesis (section 1.3) are recalled.

- Are state-of-the-art methods considered to predict PM2.5 concentrations and visibility several hours in advance?
- Can the configuration of the chosen state-of-the-art method achieve the best performance?
- Can the methodology show better performance compared with other state-of-the-art methods in terms of root mean square errors (RMSE), mean absolute errors (MAE), confusion matrices and scatter plot diagrams?

6.2.1 Research Objective 1

The first research question is as follows.

- Are state-of-the-art methods considered to predict PM2.5 concentrations and visibility several hours in advance?

According to the literature study, many efforts have been made to enrich approaches for air pollutant and visibility predictions in recent years. These approaches can be mainly divided into two categories: differential equations based and data based methods. Additionally, as a kind of data based methods, deep learning has received immense attention in both academy and industry. Numerous research and applications have been done in this area. Therefore, the scope of this research is restricted to deep learning approaches in the beginning.

Furthermore, the focus is limited to LSTM NN. The reasons are as follows. LSTM NN have been applied for PM2.5 predictions successfully and it has been

revealed that there is a strong relationship between PM2.5 and visibility. However, up to now, LSTM NN have not been tried for visibility predictions. Since our goal is to establish deep learning architectures that can forecast PM2.5 and visibility accurately, according to the above analysis, LSTM NN are very appropriate for achieving this objective.

As for the prediction interval, in the real world, PM2.5 concentrations should be predicted many hours in advance so that there is enough time for governments and environmental agencies to provide services to protect their citizens. However, with the increase of the prediction period, the performance tends to degrade. There is a trade-off between high accuracy and a long prediction period. In our proposed LSTM methodology, the goal is to predict PM2.5 12 hours in advance. Besides, this prediction length is consistent with the average prediction length of Lotos-Euros, so that the proposed framework can be evaluated by comparing with Lotos-Euros.

As mentioned at the beginning of this thesis (section1.2), visibility variations are mainly influenced by airborne particles and weather patterns. However, these two causes are hard to be determined and no mature physical or mathematical model exists for visibility. Since the visibility prediction remains a very challenging task, its prediction interval is fixed to be 4 hours, which also indeed has practical meanings because airport operators can have enough time to take measures to reduce the economic loss and passengers inconvenience caused by the air traffic disruption of low visibility.

6.2.2 Research Objective 2

The second research question is as follows.

- Can the configuration of the chosen state-of-the-art method achieve the best performance?

LSTM NN as a kind of neural networks, hyperparameters including gradient descent hyperparameters and model hyperparameters are required to be configured before training. In order to achieve the best prediction performance, different

schemes were used for different hyperparameters.

For PM2.5 predictions, the mini batch gradient descent hyperparameters were mainly investigated through the literature study. As for the model hyperparameters, the best input interval was determined by coordinate descent. Experiments showed that the granularity of the short-term data is significant enough to capture changes in air pollution and the best input interval is 18. What's more, grid searches were used for exploring the most appropriate number of hidden layers and hidden units per hidden layer. Through experiments, considering the prediction performance and training efficiency, we fixed the number of layers to be 2 and the number of neurons per layer to be 100. Additionally, feature engineering was implemented to select the most suitable input features. The effectiveness of containing nearby stations was demonstrated by comparing it with the identical configured network but discarding nearby PM2.5 data as input.

When it comes to PM10 predictions, since transfer learning was realized through model fine-tuning, the only hyperparameter that required be configured was the epoch, which was determined by comparing the training loss and validation loss in each epoch. As for the weights used in the target function of visibility predictions, a coarse-to-fine sampling scheme was adopted. All these efforts ensured that the configuration of the chosen method could achieve the best performance.

6.2.3 Research Objective 3

The third research question we discussed here is as follows.

- Can the methodology show better performance compared with other state-of-the-art methods?

For PM2.5 predictions, the performance was compared with Lotos-Euros (a regional CTMs-based system) and an LSTM-based prediction framework (Fan J et al. (2017) [9]) that also considers spatial correlations among stations and aims at predicting PM2.5 8 hours in advance in a similar region. Through analyzing their

scatter plots and confusion matrices in the half-year period, it can be concluded that the proposed spatiotemporal LSTM model overcomes the systematic underestimation that Lotos-Euros generally encounters and outperforms Lotos-Euros to some extent. Besides, the proposed framework has better performance than Fan's spatiotemporal prediction framework in terms of RMSE and MAE.

As for PM10 predictions, the training efficiency can be improved significantly by transferring knowledge from PM2.5 predictions to PM10 predictions through model fine-tuning. Besides, with the help of the scatter plots, compared with Lotos-Euros, the LSTM framework also has competitive performance in PM10 predictions.

As the first attempt at applying LSTM NN for visibility predictions, considering the 4-hour prediction period and the extremely complicated mechanism of visibility, forecasts are acceptable in practice. The total accuracy rate reaches 90.61%. The recall rate of the normal situation (L1) is 93% while its precision rate is 96%, indicating its superior prediction performance in the normal situations. Besides, for each visibility level, the number of correct predictions is larger than that of negative predictions.

6.3 Recommendations for Future Research

Future work will focus on the generalization and updates of the proposed methods. The following activities are recommended as the follow up of this thesis.

6.3.1 Prototype for Station Selections in China

In this research, experiments were just implemented in several locations. If more data are available, it will be helpful to build a prototype that could automatically collect and summarize data from the target station as well as data at nearby stations. Because air quality monitoring stations have been built in China since 2013, valid time intervals for various stations are required to be considered. Indeed, some works for framework generalization have been done. A short description is as follows.

We built a prototype that could automatically collect and complete meteorological and air quality data from the target station and PM_{2.5} data at nearby stations for all air quality monitoring stations in China. In order to include valid PM_{2.5} data at nearby stations as input, some restrictions should be considered, which are given in Table 6.1. Firstly, the searching radius to find nearby stations is limited to be 0.8 degrees. Afterward, since the beginning timestamps for different stations are not identical, the time gap is set to be three-month. That is, if the beginning timestamp of the searched nearby station is three months later than that of the target station, then this nearby station will be discarded. If the number of the remaining stations is zero, then the corresponding target station will be removed as well. If the number of the remaining stations is larger than three, then three nearby stations will be selected randomly from these remaining stations.

Table 6.1 Restrictions for searching nearby stations

Parameter	Value
Maximum Distance	0.8 degree in radius
Maximum Period Gap	3 months
Minimum of Nearby Stations	1
Maximum of Nearby Stations	3

After the selection, the distribution of monitoring stations in China is shown in the right of Figure 6.1, while the original distribution is shown in the left. According to these two pictures, most of the stations are kept, especially in the southeast China.

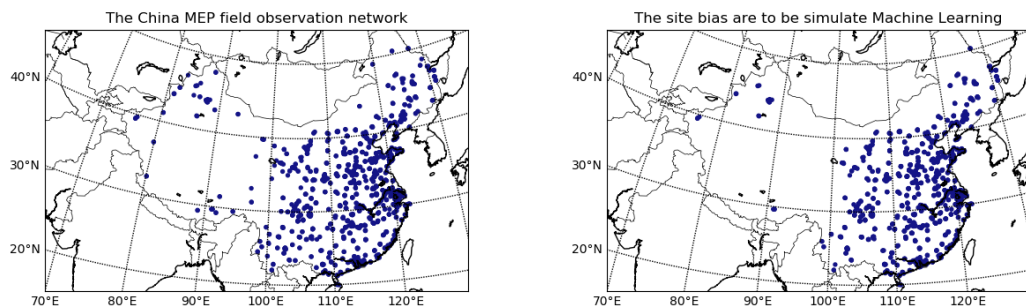


Figure 6.1 (*left*) The distribution of monitoring stations in China; (*Right*) The distribution of monitoring stations in China after selection.

6.3.2 Framework Generalization

In this thesis, for air pollutant predictions, the focus is on PM2.5 or PM10. However, the methodology proposed in this thesis is designed to be applicable for a wide range of applications; some steps of the methodology can be used to perform similar tasks such as predicting other air pollutant concentrations such as SO₂, NO₂, and O₃.

What's more, the framework output can be extended to multiple stations, timesteps or features. For example, we can output the air pollutant concentrations at all stations within a city at a time. What's more, the framework can be designed to predict the concentrations of many different air quality features at a station once. Experiments can also be done to try to predict the air pollutant concentrations in several timesteps at a time. This way, degradation possibility for much more complicated tasks should be evaluated.

6.3.3 Framework Update

In this thesis, data in the specific period were accessed. After splitting the obtained dataset into three parts, the training set is used to train the network. Once the training set is updated, the framework should be trained again. That is, the proposed method is offline. However, in the real world, meteorological and air quality data is updated hourly, online methods should be considered for application purposes

References

- [1] Feng X, Li Q, Zhu Y, et al. Artificial neural networks forecasting of PM_{2.5} pollution using air mass trajectory based geographic model and wavelet transformation[J]. *Atmospheric Environment*, 2015, 107: 118-128.
- [2] Reddy V, Yedavalli P, Mohanty S, et al. Deep Air: Forecasting Air Pollution in Beijing, China[J].
- [3] Zhao P, Zhang X, Xu X, et al. Long-term visibility trends and characteristics in the region of Beijing, Tianjin, and Hebei, China[J]. *Atmospheric Research*, 2011, 101(3): 711-718.
- [4] Zhu L, Zhu G, Han L, et al. The Application of Deep Learning in Airport Visibility Forecast[J]. *Atmospheric and Climate Sciences*, 2017, 7(03): 314.
- [5] Peng H. Air quality prediction by machine learning methods[D]. University of British Columbia, 2015.
- [6] Li X, Peng L, Yao X, et al. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation[J]. *Environmental Pollution*, 2017, 231: 997-1004.
- [7] Pedrycz W, Chen S. Time Series Analysis, Modeling and Applications[J]. *A Computational Intelligence Perspective (e-book Google)*, 2013.
- [8] Cornejo-Bueno L, Casanova-Mateo C, Sanz-Justo J, et al. Efficient Prediction of Low-Visibility Events at Airports Using Machine-Learning Regression[J]. *Boundary-Layer Meteorology*, 2017, 165(2): 349-370.
- [9] Fan J, Li Q, Hou J, et al. A spatiotemporal prediction framework for air pollution based on deep RNN[J]. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017, 4: 15.
- [10] Li X, Peng L, Hu Y, et al. Deep learning architecture for air quality predictions[J]. *Environmental Science and Pollution Research*, 2016, 23(22): 22408-22417.
- [11] Shekhar S, Jiang Z, Ali R Y, et al. Spatiotemporal data mining: a computational perspective[J]. *ISPRS International Journal of Geo-Information*, 2015, 4(4): 2306-2338.
- [12] Atluri G, Karpatne A, Kumar V. Spatio-Temporal Data Mining: A Survey of Problems and Methods[J]. *arXiv preprint arXiv:1711.04710*, 2017.
- [13] Seigneur C, Moran M. Chemical-Transport Models[J].

- [14] Manders A M M, Builtjes P J H, Curier L, et al. Curriculum vitae of the LOTOS–EUROS (v2. 0) chemistry transport model[J]. *Geosci Model Dev* 2017; 10: 4145-73, 2017.
- [15] Manders-Groot A M M, Segers A J, Jonkers S, et al. LOTOS-EUROS v2. 0 reference guide[J]. TNO report TNO2016, 2016, 10898.
- [16] Timmermans R, Kranenburg R, Manders A, et al. Source apportionment of PM2. 5 across China using LOTOS-EUROS[J]. *Atmospheric Environment*, 2017, 164: 370-386.
- [17] Van Damme M, Wichink Kruit R J, Schaap M, et al. Evaluating 4 years of atmospheric ammonia (NH₃) over Europe using IASI satellite observations and LOTOS-EUROS model results[J]. *Journal of Geophysical Research: Atmospheres*, 2014, 119(15): 9549-9566.
- [18] Chen J, Lu J, Avise J C, et al. Seasonal modeling of PM2.5 in California's San Joaquin Valley[J]. *Atmospheric environment*, 2014, 92: 182-190.
- [19] Prank M, Sofiev M, Tsyro S, et al. Evaluation of the performance of four chemical transport models in predicting the aerosol chemical composition in Europe in 2005[J]. *Atmospheric Chemistry and Physics*, 2016, 16(10): 6041-6070.
- [20] Saide P E, Carmichael G R, Spak S N, et al. Forecasting urban PM10 and PM2. 5 pollution episodes in very stable nocturnal conditions and complex terrain using WRF–Chem CO tracer model[J]. *Atmospheric Environment*, 2011, 45(16): 2769-2780.
- [21] Díaz-Robles L A, Ortega J C, Fu J S, et al. A hybrid ARIMA and artificial neural networks model to forecast particulate matter in urban areas: The case of Temuco, Chile[J]. *Atmospheric Environment*, 2008, 42(35): 8331-8340.
- [22] Osowski S, Garanty K. Forecasting of the daily meteorological pollution using wavelets and support vector machine[J]. *Engineering Applications of Artificial Intelligence*, 2007, 20(6): 745-755.
- [23] Hoi K I, Yuen K V, Mok K M. Kalman filter based prediction system for wintertime PM10 concentrations in Macau[J]. *Global NEST Journal*, 2008, 10(2): 140-150.
- [24] Sun W, Zhang H, Palazoglu A, et al. Prediction of 24-hour-average PM2. 5 concentrations using a hidden Markov model with different emission distributions in Northern California[J]. *Science of the total environment*, 2013, 443: 93-103.
- [25] Ong B T, Sugiura K, Zettsu K. Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2. 5[J]. *Neural Computing and Applications*, 2016, 27(6): 1553-1566.

- [26] Li T, Shen H, Zeng C, et al. Point-surface fusion of station measurements and satellite observations for mapping PM_{2.5} distribution in China: Methods and assessment[J]. *Atmospheric Environment*, 2017, 152: 477-489.
- [27] Goodfellow I, Bengio Y, Courville A, et al. *Deep learning*[M]. Cambridge: MIT press, 2016.
- [28] Li T, Shen H, Yuan Q, et al. Estimating Ground-Level PM_{2.5} by Fusing Satellite and Station Observations: A Geo-Intelligent Deep Learning Approach[J]. *Geophysical Research Letters*, 2017, 44(23).
- [29] Biancofiore F, Busilacchio M, Verdecchia M, et al. Recursive neural network model for analysis and forecast of PM₁₀ and PM_{2.5}[J]. *Atmospheric Pollution Research*, 2017, 8(4): 652-659.
- [30] Hochreiter S, Schmidhuber J. Long short-term memory[J]. *Neural computation*, 1997, 9(8): 1735-1780.
- [31] Ghaderi A, Sanandaji B M, Ghaderi F. Deep forecast: deep learning-based spatio-temporal forecasting[J]. *arXiv preprint arXiv:1707.08110*, 2017.
- [32] Zhang Q, Wang H, Dong J, et al. Prediction of sea surface temperature using long short-term memory[J]. *IEEE Geoscience and Remote Sensing Letters*, 2017, 14(10): 1745-1749.
- [33] Zaytar M A, El Amrani C. Sequence to sequence weather forecasting with long short term memory recurrent neural networks[J]. *Int J Comput Appl*, 2016, 143(11).
- [34] Prakash A, Kumar U, Kumar K, et al. A wavelet-based neural network model to predict ambient air pollutants' concentration[J]. *Environmental Modeling & Assessment*, 2011, 16(5): 503-517.
- [35] Dutta D, Chaudhuri S. Nowcasting visibility during wintertime fog over the airport of a metropolis of India: decision tree algorithm and artificial neural network approach[J]. *Natural Hazards*, 2015, 75(2): 1349-1368.
- [36] Fan G, Ma H, Zhang X, et al. Impacts of relative humidity and PM_{2.5} concentration on atmospheric visibility: A comparative study of hourly observations of multiple stations. *Acta Meteorologica Sinica*, 2016, 74(6): 959-973
- [37] Wang X, Han J, Chen J, et al. Variation Characteristics of Atmospheric Visibility and Their Relationship with Relative Humidity and Particle Concentration in Shijiazhuang of Hebei[J]. *Journal of Arid Meteorology*, 2016, 34(4): 648-655.
- [38] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. *The Journal of Machine Learning Research*, 2014, 15(1):

1929-1958.

- [39] Gal Y, Ghahramani Z. A theoretically grounded application of dropout in recurrent neural networks[C]//Advances in neural information processing systems. 2016: 1019-1027.
- [40] Ruder S. An overview of gradient descent optimization algorithms[J]. arXiv preprint arXiv:1609.04747, 2016.
- [41] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
- [42] Cao Z, Yu S, Xu G, et al. Multiple adaptive kernel size KLMS for Beijing PM2. 5 prediction[C]//IJCNN. 2016: 1403-1407.
- [43] Zhang S. Nearest neighbor selection for iteratively kNN imputation[J]. Journal of Systems and Software, 2012, 85(11): 2541-2552.
- [44] Keskar N S, Mudigere D, Nocedal J, et al. On large-batch training for deep learning: Generalization gap and sharp minima[J]. arXiv preprint arXiv:1609.04836, 2016.