Deep Learning models for Spatial Prediction of Fine Particulate Matter

Luis Ernesto Colchado Soncco

Advisor: PhD. Jose Eduardo Ochoa Luna

Committee Members:

PhD. Alex Jesús Cuadros Vargas – Universidad Católica San Pablo – Perú

PhD. Pablo Cesar Calcina Ccori – Fundação Getulio Vargas – Brasil

PhD. Edwin Rafael Villanueva Talavera – Pontificia Universidad Católica del Perú – Perú

Thesis submitted to the
Department of Computer Science
in partial fulfillment of the requirements for the degree of
Master in Computer Science.

Universidad Católica San Pablo – UCSP
November of 2023 – Arequipa – Peru

Tesis Maestría Ciencia Computación (version final)

INFORME DE ORIGINALIDAD

14%	6%	12%	%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Luis E. Colchado, Edwin Villanueva, Jose Ochoa-Luna. "A Neural Network Architecture with an Attention-based Layer for Spatial Prediction of Fine Particulate Matter", 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), 2021 Publicación	9%
2	repositorio.ucsp.edu.pe Fuente de Internet	2%
3	link.springer.com Fuente de Internet	1%
4	www.slidestalk.com Fuente de Internet	1%
5	www.researchgate.net Fuente de Internet	1%
6	"Machine Learning in Chemical Safety and Health", Wiley, 2022 Publicación	1%

To my parents, Luis and Lourdes, who never stop giving themselves in countless ways. To my brother Gabriel, who is one more motivation to get ahead.

Abbreviations

CETESB Companhia Ambiental do Estado de São Paulo

AQM Air Quality Monitoring

$PM_{2.5}*$ Synthetic Fine Particulate Matter

$PM_{2.5}$ Particulate Matter less than 2.5 micrometers in diameter

PM_{10} Particulate Matter less than 10 micrometers in diameter

CO Carbon Monoxide

CO_2 Carbon Dioxide

O_3 Ozone

SO Sulfur Monoxide

SO_2 Sulfur Dioxide

NO_2 Nitrogen Dioxide

NO Nitrogen Monoxide

T Temperature

P Pressure

RH Relative Humidity

DP Dew Point

WD Wind Direction

WS Wind Speed

LAADS DAAC Level-1 and Atmosphere Archive and Distribution System Distributed Active Archive Center

AOD Aerosol Optical Depth

VIIRS Visible Infrared Imaging Radiometer Suite

USGS United States Geological Survey

DEM Digital Elevation Model

SRTM Shuttle Radar Topography Mission

NDVI Normalized Difference Vegetation Index

NTL Nighttime Lights

DNB Day/Night Band

CMAQ Community Multiscale Air Quality Modeling System

WRF-Chem Weather Research and Forecasting model coupled with Chemistry

DAL Deep Air Learning

AQI Air Quality Index

ARIMA Autoregressive Integrated Moving Average

SVM Support Vector Machine

MLP Multilayer Perceptron

MLR Multiple Linear Regression

RBFNN Radial Basis Function Neural Network

RNN Recurrent Neural Network

STDL Spatiotemporal Deep Learning

LSTM NN Long Short-Term Memory Neural Network

SVR Support Vector Regression

TDNN Time Delay Neural Network

EPA Environmental Protection Agency

ANN Artificial Neural Networks

FCNN Fully Connected Neural Network

FCL2 Fully Connected Neural Network with L2 loss

FCSL Fully Connected Neural Network with spatial loss

ARMA Autoregressive Moving Average

SAE Stacked Autoencoder

STANN Spatiotemporal Artificial Neural Network

LSTM Long Short Term Memory

LSTME Long Short Term Memory Extension

***k*NN** k-Nearest Neighbor

GAT Graph Attention Network

AffinityNet Affinity Network

BP Backpropagation

BPTL Backpropagation Through Time

LR Logistic Regression

PDF Probability Density Function

G Generative Network

D Discriminative Network

GAN Generative Adversarial Network

cGAN Conditional Generative Adversarial Network

z Latent Space

SLoss Spatial Loss

cGANSL Spatial-learning Conditional Generative Adversarial Network

RMSE Root Mean Square Error

R2 Coefficient of determination

MAE Mean Absolute Error

IDW Inverse Distance Weighting

OK Ordinary Kriging

Acknowledgments

First and foremost, I want to thank God for having guided me throughout these years of study.

I want to thank my parents, brother, and my entire family for their continued support and encouragement during these years of study.

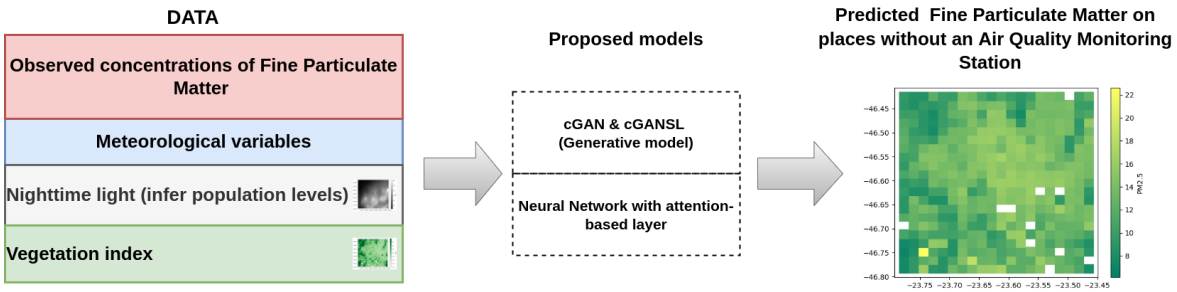
I want to thank in a particular way to the Universidad Católica San Pablo (UCSP) have allowed the grant and financing my studies in the Master Program in Computer Science. Likewise, I also want to acknowledge and appreciate the financial support of the Concytec - World Bank project "Mejoramiento y Ampliación de los Servicios del Sistema Nacional de Ciencia, Tecnología e Innovación Tecnológica" 8682-PE, through its executing unit ProCiencia [contract 50-2018-FONDECYT-BM-IADT-MU].

Moreover, I would like to express my gratitude and appreciation to my advisor Jose Ochoa, well as my co-advisor Edwin Villanueva for giving his guidance and support throughout the Master's program and the preparation of this thesis.

I also would like to take this opportunity to say warm thanks to all my beloved friends, who have been so supportive along the way of doing this thesis.

Finally, I thank all the people who directly or indirectly helped me in the preparation and presentation of this work, either by exchanging ideas, giving me advice and recommendations or by encouraging me to continue.

Abstract



Studies indicate that air pollutant concentrations affect human health. Especially, Fine Particulate Matter ($PM_{2.5}$) is the most dangerous pollutant because this is related to cardiovascular and respiratory diseases, among others. Therefore, governments must monitor and control pollutant concentrations. To this end, many of them have implemented Air quality monitoring (AQM) networks. However, AQM stations are usually spatially sparse due to their high costs in implementation and maintenance, leaving large areas without a measure of pollution. Numerical models based on the simulation of diffusion and reaction process of air pollutants have been proposed to infer their spatial distribution. However, these models often require an extensive inventory of data and variables, as well as high-end computing hardware. In this research, we propose two deep learning models. The first is a generative model called Conditional Generative adversarial Network (cGAN). Additionally, we add a loss based on the predicted observation and the k nearest neighbor stations to smooth the randomness of adversarial learning. This variation is called Spatial-learning cGAN (cGANSL), which got better performance for spatial prediction. To interpolate $PM_{2.5}$ on a location, cGANSL and classical methods like Inverse Distance Weighting (IDW) need to select the k nearest neighbor stations based on straight distance. However, this selection may leave out data from more distant neighbors that could provide valuable information. In this sense, the second proposed model in this study is a Neural Network with an attention-based layer. This model uses a recently proposed attention layer to build a structured graph of the AQM stations, where each station is a graph node to weight the k nearest neighbors for nodes based on attention kernels. The learned attention

layer can generate a transformed feature representation for unobserved location, which is further processed by a neural network to infer the pollutant concentration. Based on data from AQM network in Beijing, meteorological conditions, and information from satellite products such as vegetation index (NDVI) and human activity or population-based on Nighttime Light product (NTL). The cGANSL had a better performance than IDW, Ordinary Kriging (OK), and Neural Network with an attention mechanism. In this experiment, spatial prediction models that selected the k nearest neighbors had a good performance. That may be AQM station Beijing's high correlation between them. However, using data from the AQM network of São Paulo, where AQM stations have a low correlation, the Neural network with an attention-based layer have better performance than IDW, OK, and cGANSL. Besides, the normalized attention weights computed by our attention model showed that in some cases, the attention given to the nearest nodes is independent of their spatial distances. Therefore, the attention model is more flexible since it can learn to interpolate $PM_{2.5}$ concentration levels based on the available data of the AQM network and some context information. Finally, we found that NDVI and NTL are high related to air pollutant concentration predicted by the attention model.

Keywords: Spatial prediction, Fine particulate matter, k-Nearest Neighbors, Generative modeling, Attention mechanism, Deep Learning

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In Section 1.1 we describe the motivation and context of our work and Section 1.2 presents our problem statement. Section 1.3 shows the objectives of this work. Section 1.4 shows the contributions of this work and Section 1.5 describes the structure of this thesis document.

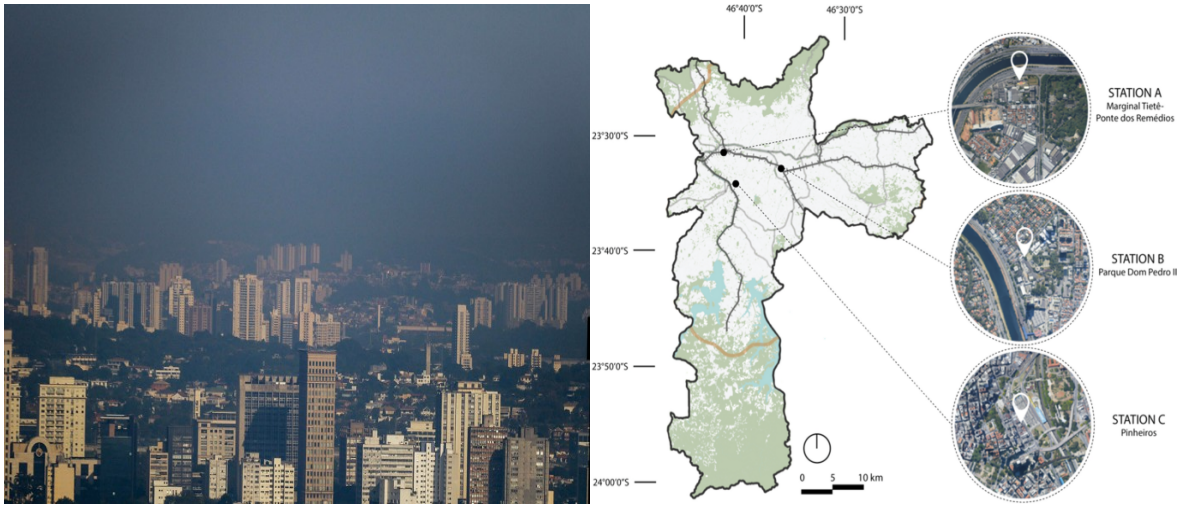
1.1 Motivation and Context

Nowadays, air pollution is a critical problem due to its effect on human health. The most common harmful air pollutants are Carbon Monoxide (CO), Carbon Dioxide (CO_2), Sulfur Monoxide (SO), Sulfur Dioxide (SO_2), Nitrogen Monoxide (NO), Nitrogen Dioxide (NO_2), and Ozone (O_3), which cause diseases especially in children, because they are more exposed to air pollution and their organs are still developing (Braga et al., 1999; Najjar, 2011). Other harmful pollutants are Particulate Matter less than 2.5 micrometers in diameter ($PM_{2.5}$) and Particulate Matter less than 10 micrometers in diameter (PM_{10}), due to the fine particles that can enter the human body through the respiratory tract (Pražnikar and Pražnikar, 2012; Du et al., 2015). In this regard, the Environmental Protection Agency (EPA) states that epidemiological evidence provides coherence and biological plausibility between these air pollutants and cardiovascular mortality in older adults and respiratory diseases in children (Agency, 2009).

Therefore, many governments have been implementing Air Quality Monitoring (AQM) stations and control projects to improve the decision-making process of organizations responsible for the environment and the health of the population. These organizations know the pollution levels in the regions of the AQM stations according to the Air Quality Index (AQI) standard that is calculated based on six pollutant concentrations observed for each station: (O_3, $PM_{2.5}$, PM_{10}, CO, SO_2 and NO_2) (Mintz, 2009).

China is one of the countries with the highest AQI in the world due to the high industrial and vehicular emissions that increase the levels of air pollutant concentrations (Guan et al., 2016). In this regard, in February 2012, China's Ministry of Environmental Protection launched a new standard of environmental air quality to set the new allowed AQI limit values. However, despite these limit values, Lowsen and Conway (2016) states that 28.5% of Beijing's days exceed the Chinese air quality standards in 2015, according to the information collected by the AQM stations, implemented in that city.

In South America, São Paulo is a metropolitan area with a population of 21 million and this is one of the cities with major air pollutant levels (Figure 1.1a). Therefore, Companhia Ambiental do Estado de São Paulo (CETESB) was created in 1970, a fact that motivated the beginning of studies on air quality, based on pollution data collected by AQM networks with stations implemented in different points by this organization in 1975 (de Fatima Andrade et al., 2017). Figure 1.1b shows three station of the air quality network of São Paulo.



(a) (b)

Figure 1.1: (a) Air pollution in São Paulo. (b) Air quality monitoring stations in São Paulo. Source: (Noda et al., 2021).

However, such AQM station networks are usually composed of few stations due to their high implementation and maintenance costs (Chow, 1995; Taborda et al., 2020). Researchers have developed air quality models to complement AQM networks and infer air pollution levels across a geographic region (Zhang et al., 2018a). In this regard, there are two main types of methods to do so: deterministic and statistical.

The deterministic methods are based on physical meteorology and chemical models to simulate the diffusion, dispersion, and interaction processes among pollutant species (Saide et al., 2011; Chen et al., 2014). However, they are complex to implement and the collection of data is more difficult because they require a large number of variables to model the dispersion of the pollutants in the atmosphere (Kumar and Goyal, 2011). Additionally, these models usually need highly precise emission inventory

data and a high-end computing platform (Delavar et al., 2019).

On the other hand, statistical methods are data-driven methods that estimate air pollutant concentrations without building complex models of the physical-chemical process. Among the most popular traditional statistic models for interpolation are (Inverse Distance Weighting (IDW) and Ordinary Kriging (OK)) (Wong et al., 2004; Li et al., 2014; Masroor et al., 2020). They infer values at unobserved points on the map based on the weighting of the values observed by AQM stations.

The results of simple statistical models are usually unsatisfactory for predicting air pollutant levels in urban environments due to the emission and dispersion of fine particulate matter is a complex and dynamic process, affected by different factors such as meteorological conditions, anthropomorphic activities, and land-related variables (Zheng et al., 2013; Wei et al., 2019; Wang et al., 2019; Vargas-Campos and Villanueva, 2021). This has motivated new modeling approaches based on deep learning techniques. The main characteristic of deep learning models is their ability to capture complex concepts from simpler concepts in a hierarchical way (Goodfellow et al., 2016).

Models based on deep learning have been proposed in the air quality field to estimate unobserved air pollutant concentrations over space and time. Some studies combined deep learning models with statistical interpolation models to predict over space and time by a hybrid model, capturing spatio-temporal dependencies of the data on air pollution process (Guo et al., 2020; Ma et al., 2019; Fan et al., 2017). In addition, models to predict and interpolate air pollutant concentrations by a single model based on semi-supervised learning focused on a spatio-temporal correlation (Qi et al., 2018).

There is another deep learning approach called Generative modeling. Generative models are trained to learn some hidden or underlying structure of input data to represent probability distributions of it. Generative Adversarial Network (GAN) (Goodfellow et al., 2014) is a generative model, which is adjusted based on an adversarial learning algorithm, where there is the competition of two Artificial Neural Networks (ANN), which are Generative Network (G) and Discriminative Network (D). G model aims to represent the underlying distribution of the sample data to generate synthetic data similar to real data, while D model aims to discriminate real from synthetic data accurately. This architecture shows satisfactory performance on many real-world problems (Gui et al., 2020; Alqahtani et al., 2021; Jabbar et al., 2021). In that sense, studies proposed this method to impute generated synthetic data at unobserved points (Friedjungová et al., 2020). GAN does not exert control over the generated data, producing data only from a Latent Space (z). However, adding conditional information can direct the generation process to generate data for different modalities or based on different class labels.

In that sense, Mirza and Osindero (2014) proposed an extension of GAN called Conditional Generative Adversarial Network (cGAN). cGAN add conditional input variables on G and D making the models output be based on class labels. This variant has been applied to many topics, especially on computer vision, where for example a novel architecture was proposed capable of doing image-to-image translation (Isola

et al., 2017). In the context of air quality, currently, cGAN was only proposed in the task of forecasting air pollution concentrations (Toutouh, 2021).

In this work, we propose two approaches. The first one is the application of cGAN to predict $PM_{2.5}$ concentration values at unobserved points. In this model, the generated values are conditioned by meteorological conditions, variables related to earth and population, as well as k-Nearest Neighbor (kNN) information. These variables were chosen because they are informative of the local context of the possible levels of contamination and because they can reduce the noise of cGAN predictions. We embed a spatial loss to smooth adversarial loss. This variation was called Spatial-learning Conditional Generative Adversarial Network (cGANSL). The spatial loss is a kind of weighting loss that allows the error calculation for an unlabeled point based on the spatial distance to the AQM stations and their observed values.

The second approach is the application of a recently proposed kNN attention polling layer (Ma and Zhang, 2019) to the domain of air quality. Most statistical and machine learning models (like cGANSL) for spatial prediction construct their predictions based on station distances. Commonly, they identify the kNN stations that provide the most predictive information to infer at the unobserved point. The suitable k is usually found by experimenting. The most common distance metric used to assess station proximity is the straight-line distance. However, in the domain of air pollution, the straight-line distance can not be the best option to estimate at unobserved points. Environmental factors such as weather, anthropomorphic activity, and topographic or geographical variables can affect the actual importance of the station observations to the prediction point. Therefore, we propose the application of the kNN attention polling layer, which uses a self-attention strategy to calculate how much attention each node should pay to neighboring nodes in the structured graph of the data. This approach is not biased by the election of the proximity function (i.e. straight-line distance), but the such function is implicitly learned from data. Thus, the attention-based layer creates a transformed feature representation for each node, which is further processed by a ANN model to infer $PM_{2.5}$ concentrations at unobserved points.

On an extensive set of experiments with the proposed and standard approaches on Beijing and São Paulo AQM networks showed that cGANSL is more accurate on Beijing data than the neural network with kNN attention polling layer, IDW, OK and cGAN without a spatial loss, based on three metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Coefficient of determination (R2). However, the results on São Paulo data showed that the neural network with kNN attention polling layer outperforms the performance of the other models (cGANSL, IDW and OK) according to RMSE, MAE, and R2 metrics.

Furthermore, we found that observations of pollution concentrations on AQM stations in Beijing are more spatially correlated than in São Paulo city. This means that the spatial distance is more relevant in Beijing than in São Paulo. This fact may explain our finding that the models that use selected kNN stations by spatial distance (cGANSL) had better results compared to the neural network based on the attention layer. Instead, in the São Paulo network, the AQM stations have low correlations

among them, which gives the opportunity to the attention mechanism to better exploit such information. In fact, we found in the São Paulo model, that the normalized attention weights computed by kNN attention polling layer were independent of the straight-line distance.

Finally, we found a relationship between the predicted values of air pollutant concentrations and information related to earth and population, which are Normalized Difference Vegetation Index (NDVI) and Nighttime Lights (NTL), respectively.

1.2 Problem Statement

Predicting particulate matter concentrations at unknown locations is of great importance to get knowledge about the air quality distribution in urban areas and to inform authorities to improve it. However, this task is highly challenging due to the dynamism and complexity of atmospheric processes, pollutant emissions and urban environments.

Various numerical and statistical models have been proposed to deal with this problem. However, numerical methods often require high computing power and emission inventory information, which is difficult to obtain in practice. On other hand, statistical methods make very simplistic assumptions about pollution processes, giving frequently unsatisfactory results.

Recently, generative models have been applied to several problems to capture the underlying data distribution and generate synthetic data. However, the potential of these models has been little explored for the inference of pollution concentrations in unobserved locations. Generative models could be appropriate to learn the distribution of the air pollutant concentrations measured by the monitoring stations and thus be able to infer the possible concentrations in new locations. Furthermore, recent advances in these models would allow inference to be conditioned on local variables such as meteorology and urban variables. However, it is still unknown how advantageous such an application would be in practice.

Another limitation identified in current machine learning and statistical models for spatial prediction is the selection of the kNN stations to construct the prediction. Usually, the stations are selected based on the straight-line distances to the prediction point. However, using the straight-line distance as a way to ponder the station's importance does not always ensure an optimal prediction since the importance of the stations can also be affected by environmental and land-use variables. Recently, a new kind of neural network layer called kNN attention polling layer, has been proposed to encode spatial correlation structure based on data. However, the potential of such an approach has not been tested yet in air quality prediction.

Therefore, our main research questions are:

- Can a generative model conditioned by meteorological variables and kNN in-

formation predict the $PM_{2.5}$ concentrations at unknown locations with better performance than traditional interpolation models?

- Can the new kNN attention polling layer improve the spatial prediction of $PM_{2.5}$ concentration with respect to using the traditional straight-line distance as a way to weighting observed values of AQM stations?

1.3 Objectives

General Objective

Our main objective is to propose a method for the spatial prediction of $PM_{2.5}$ concentrations based on generative modeling conditioned by meteorological variables and kNN information. Furthermore, to propose a model based on an attention mechanism to avoid selecting kNN stations by spatial distance to predict $PM_{2.5}$ concentrations more accurately.

Specific Objectives

To achieve our main objective, we have the following specific objectives:

- To develop a cGAN to predict $PM_{2.5}$ concentrations at unobserved points using meteorological information, geographic factors and $PM_{2.5}$ concentrations observed by their kNN stations.
- To develop a Neural Network with kNN Attention Polling Layer to weigh the concentrations observed by the kNN stations regardless of their selection based on the straight-line distance in order to predict $PM_{2.5}$ concentrations at unobserved points accurately.

1.4 Contributions

Basically, we propose two models for spatial prediction of $PM_{2.5}$ in this thesis. Contributions can be separated into two parts and are detailed below:

- Conditional Generative Adversarial Network and Spatial Learning to predict $PM_{2.5}$ at unobserved points.
  - We propose a generative model to predict air pollutant concentrations at points without measurements of pollution more accurately than traditional interpolation models.

  - We add a spatial loss to adversarial learning of cGAN to improve the fake values generated of $PM_{2.5}$. In order to get a better spatial prediction model.

- A Neural Network Architecture with an Attention-based Layer for Spatial Prediction of Fine Particulate Matter

  - This proposal was published in the 8th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2021) (Colchado et al., 2021).
  - We propose a kNN attention polling layer and artificial neural network to predict $PM_{2.5}$ concentrations at points without measurements of pollution more accurately than traditional interpolation models and cGAN proposed as the first method in this thesis.
  - We proposed a model using an attention-based layer to weighting kNN stations founding that the attention given to the nearest stations is independent of their spatial distances.
  - We found a high relationship between $PM_{2.5}$ predicted by our model and land-related variable, as well as population-related variable represented by NDVI and NTL respectively.

1.5 Outline

This thesis document is divided into eight chapters. After this introduction, problem formulation and objectives, Chapter 2 describes the main concepts about $PM_{2.5}$ and prediction of air pollutant concentrations, spatial interpolation models based on statistics method, as well as Deep Learning models used in this topic. Moreover, we described research studies about the prediction of air pollutant concentrations in Chapter 3. Chapter 4 shows the pre-processing and matching performed to get input data for the spatial prediction models implemented in this research. Moreover, Chapter 5 shows the first proposed method which is cGAN and additionally with a Spatial Learning approach (cGANSL). While, Chapter 6 presents the second proposed method which is a Neural Network with an attention-based layer for spatial prediction of $PM_{2.5}$. Chapter 7 and Chapter 8 show the results of the experiment of all spatial prediction models using Beijing data and São Paulo data, respectively. Finally, we present the discussion and conclusion of this work in Chapter 9.

Chapter 2

Background

This chapter presents the main definitions of the spatial predictions of $PM_{2.5}$ concentrations using statistical and deep learning models based on meteorological conditions, as well as land-related variables and population variables obtained from satellite products.

In this sense, Section 2.1 describes the $PM_{2.5}$. Commonly, to predict spatially $PM_{2.5}$ concentrations we can use traditional interpolation methods based on spatio-temporal statistics explained in Section 2.2. Moreover, we show some deep learning models that have been proposed to solve this challenge in Section 2.3.

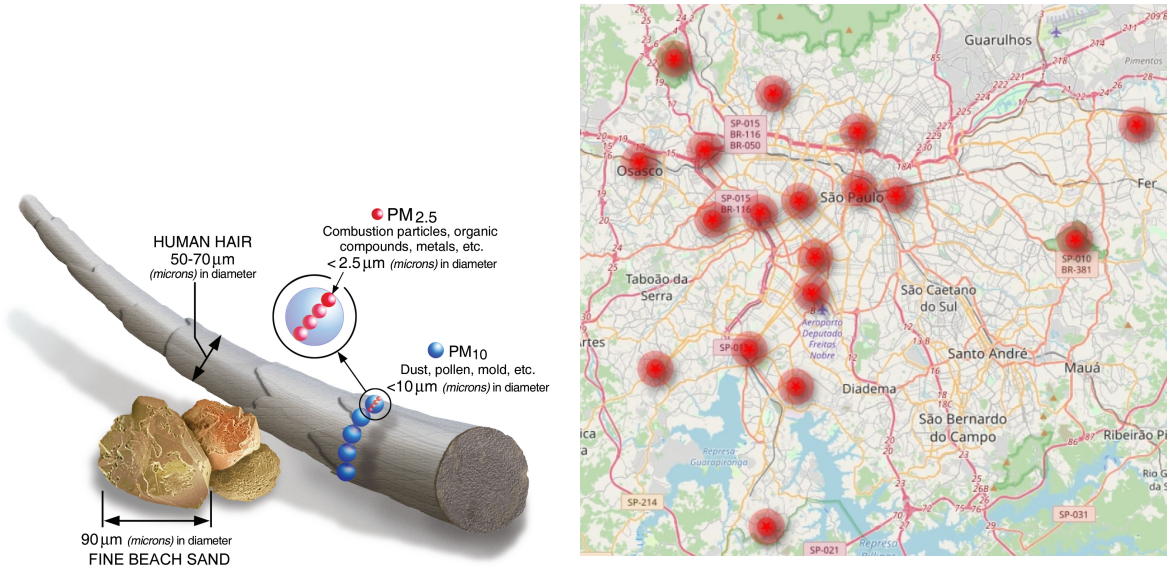
Finally, we explain land-related variables in Section 2.4 and population-related variables in Section 2.5 used in this research.

2.1 Particle Matter of less than 2.5 micrometers in diameter

The $PM_{2.5}$ is a common air pollutant, which can penetrate deep regions of the lungs due to its tiny size (Organization, 2013; Pražnikar and Pražnikar, 2012; Du et al., 2015). Figure 2.1a shows the comparison between this pollutant of 2.5 micrometers and 10 micrometers, human hair and fine beach sand.

In order to monitor air quality, based on the AQI, governments implement AQM stations, which monitor air pollutant concentrations. However, they are usually available only in a few locations due to their high cost of implementation and maintenance. Figure 2.1b shows AQM stations implemented in São Paulo covering only a few areas of the city. Each AQM station obtains a time series of $PM_{2.5}$ concentration levels ($\mu g/m^3$).

Time series is a sequence of values obtained over time intervals, which can be hourly, daily, annual and so on. Figure 2.2 shows a time series of $PM_{2.5}$ observed



(a) (b) Section

Figure 2.1: (a) Size of particulate matter of 2.5 micrometers. Source: (Agency, 2009). (b) Air quality monitoring stations are unevenly distributed in São Paulo. Source: own.

by two AQM stations and gray boxes denoting intervals with missing values, due to problems in their sensors.

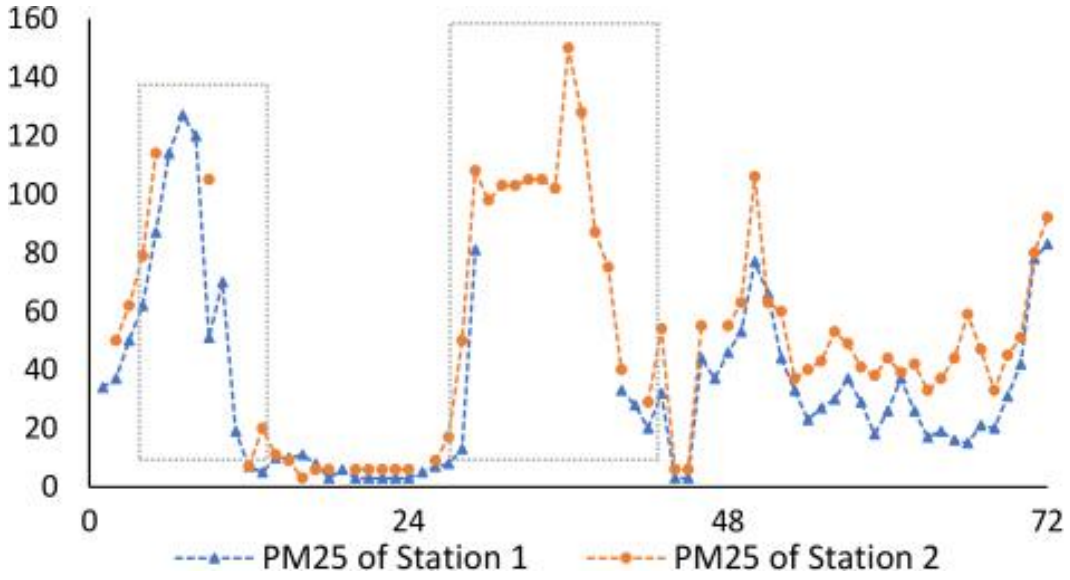


Figure 2.2: Time series of fine particulate matter and missing values. Source: (Tang et al., 2020).

2.2 Spatiotemporal statistics

Physically speaking, when we collect data to explain a phenomenon, this data usually includes space and time information, because we need to know the ”where” and the ”when” for the data being collected so that we can know the ”why” of a phenomenon. Cressie and Wikle (2011) state causation is the “holy grail” of Science, and hence to infer cause–effect relationships (i.e., “why”) it is essential to keep track of “when” a cause always precedes an effect. Keeping track of “where” recognizes the importance of knowing the “lay of the land” and, quite simply, there would be no History without geography.

Thus, we are trying to describe the expansion or contraction of different variables, where there is a complex system of physical, biological, and social processes that interact through spatiotemporal scales (Wikle et al., 2019).

Therefore, there is an essential statistical characteristic of spatio-temporal data that is very common, which is that nearby observations in space and time tend to be more alike than those far apart (Cressie and Wikle, 2011). Thus, many interpolation and prediction models are based on this characteristic.

2.2.1 Spatial Interpolation Methods

Spatial interpolation methods infer values in unobserved points from known observed values. Basically, they perform the prediction by a linear combination of the known data, given as Equation 2.1 (Longley, 2005).

$$\hat{y}_i = \sum_{j=1}^{n} \lambda_j y_j \tag{2.1}$$

where $\hat{y}_i$ represent the predicted value at point i, y_j is the value in sample point j, λ_j is the weight given to sample value j and n is the number of sample points used in the prediction of value i.

2.2.1.1 Inverse Distance Weighted (IDW)

IDW is a deterministic method that interpolates unknown points based on a weighted average of k values of known near neighbor points and the inverse distance to these points. Therefore, each unknown point is estimated according to Equation 2.2.

$$\hat{y}_i = \frac{\sum_{j=1}^{k} \frac{y_k}{d_{ij}^p}}{\sum_{j=1}^{k} \frac{1}{d_{ij}^p}} \forall i \in I \tag{2.2}$$

where $\hat{y}_i$ is the interpolated value for the instance i, d_{ij} is the distance of the point in the instance i to the neighbor j, k is the number of near neighbors and I the number

of unknown points. Moreover, p is a parameter that indicates how fast the weight of the points tends to zero with increasing distance from the unknown point (typically $p = 2$) (Schloeder et al., 2001).

2.2.1.2 Ordinary Kriging (OK)

OK differs from IDW in the way they choose the weights. OK solves the estimation problem focused on a continuous model of stochastic spatial variation creating the variograms and the covariance functions to calculate the spatial autocorrelation values (Webster et al., 2007). An empirical variogram is defined by Equation 2.3, which expresses the degree of similarity between two observations separated by a given distance (h) (Wong et al., 2004).

$$\gamma(h) = \frac{1}{2M(h)} \sum_{j=0}^{M(h)} \{z(x_i) - z(x_i + h)\}^2 \tag{2.3}$$

where $\gamma(h)$ is the estimated semivariance at a separation distance h, $z(x_i)$ and $z(x_i+h)$ are the observed values at x_i and x_i+h separated by h, of which there are $M(h)$ pairs. Then, OK uses the computed variogram to calculate the values of λ_j in order to estimate the unobserved values by Equation 2.1 minimizing the error variance of the model.

2.3 Deep Learning

The dispersion of air pollutant concentrations is affected by many factors, such as meteorological conditions, geographical features and traffic flow. Therefore, representation learning becomes a complex task because it can be difficult to learn to represent abstract and high-level factors through traditional interpolation methods. Deep Learning solves this because it allows the computer to build complex concepts from simple concepts (Goodfellow et al., 2016).

Furthermore, 'deep learning' arises from the total length of the chain, which defines the depth of a ANN. ANN is associated with a directed acyclic graph that can have many functions connected to form a chain. For example, for three functions $(f^{(1)}, f^{(2)}, f^{(3)})$, then this forms a chain $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$, where x, $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$ are input, input layer, second layer and an output layer of the ANN, respectively (Goodfellow et al., 2016). This model is detailed in the following subsection.

2.3.1 Artificial Neural Networks

ANN is a computational model, which has interconnected neurons. Neurons are elements that have an activation level. That level is variable and depends on calculating

the activation function or the state transition function and the received signals, which have an associated synaptic weight. Figure 2.3 shows an artificial neuron, where the value of the neuron's output is given as Equation 2.4.

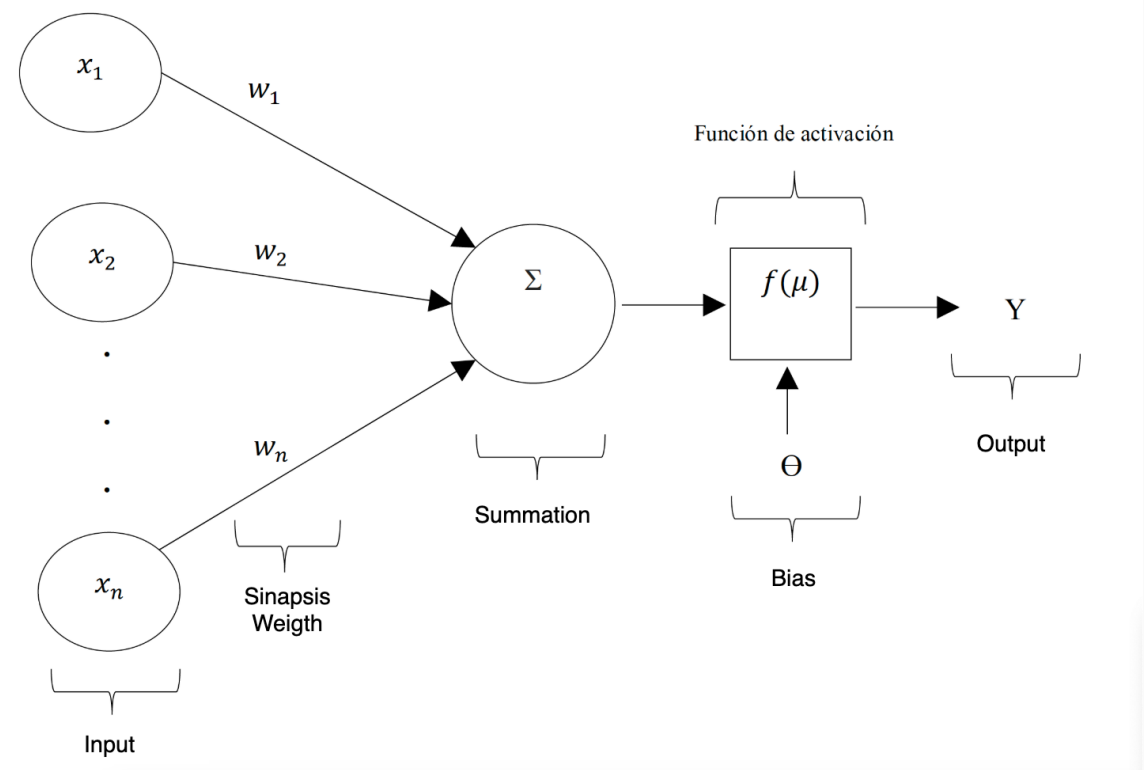


Figure 2.3: Artificial Neuron. source: Own.

$$Y = f(\sum_{i=1}^{N} x_i w_i + \theta) \tag{2.4}$$

where x_i and w_i are the input value and weight associated with the $i-th$ feature of input data, respectively. θ is the bias of the neuron.

Therefore, a neuron receives several inputs $x_1, x_2, x_3, ...$, which are multiplied for synaptic weights $w_1, w_2, w_3, ...$, the result of this weighted multiplication is further processed by the activation function, then the result is the output of a neuron, this is represented by Figure 2.3.

The output neuron is sent to the neurons interconnected to it, many interconnected neurons form an ANN.

2.3.1.1 Fully Connected Neural Networks

Fully Connected Neural Network (FCNN) is a ANN with many hidden layers and each one with many neurons. The neurons of a layer are connected to all the neurons of the

next layer. Figure 2.4 shows a FCNN with an input layer of 6 nodes, 2 hidden layers of 7 and 6 nodes or neurons in these layers, and an output layer of a single output neuron. Moreover, each layer interconnection has a weight matrix ($W[n_{l-1}, n_l]^{(l)}$) and a bias associated $\theta^{(l)}$, where, n_l indicated the number of neurons in layer l.

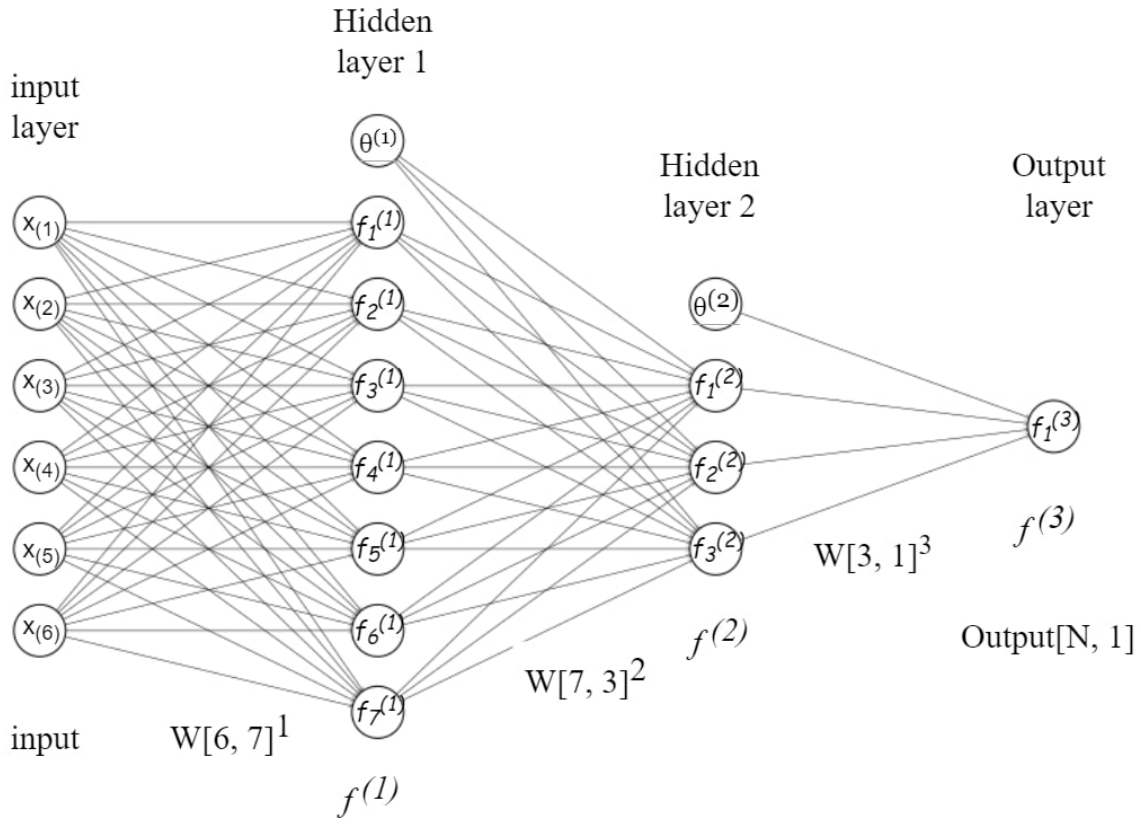


Figure 2.4: Fully Connected Neural Network (FCNN). Source: Own.

The input layer receives the signals from the exterior, which are propagated to the following hidden layers until the last of them propagates the information to the output layer, where there are one or many output neurons. This process is called Forward propagation.

After forward propagation, based on the activation of the output neuron and true value desired we can obtain output error through $L1$ (Equation 2.5) or $L2$ (Equation 2.6) loss function.

$$\mathcal{L}_{L1} = \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2.5}$$

$$\mathcal{L}_{L2} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.6}$$

where N is the number of instances in the data, y_i is desired value and $\hat{y}_i$ is the output of the FCNN for instance i.

According to the output error calculated, the ANN is adjusted using the Backpropagation (BP) algorithm (Rumelhart et al., 1986). Basically, in the first step, BP retro-propagate output error to neurons of previous layers. Then, when all neurons have their associated errors. BP algorithm updates the network's parameters based on these errors. Therefore, for the update, W is using Equation 2.7 and for θ is using Equation 2.8.

$$W^{(l)} = W^{(l)} - \alpha \frac{\partial \mathcal{L}}{W^{(l)}} \tag{2.7}$$

$$\theta^{(l)} = \theta^{(l)} - \alpha \frac{\partial \mathcal{L}}{\theta^{(l)}} \tag{2.8}$$

This process is repeated depending on the number of learning epochs that we have established.

Besides, in this research, we considered a spatial loss, which is calculated based on the kNN values, defined as Equation 2.9.

$$\mathcal{L}_{SL} = \sum_{i=1}^{N} \sum_{j=1}^{k} |y_j - \hat{y}_i| \tag{2.9}$$

where k is the number of nearest neighbors, y_j is the observed value in instance j and $\hat{y}_i$ is the estimated value by FCNN.

FCNN with $L2$ loss ($\mathcal{L}_{L2}$) we called Fully Connected Neural Network with L2 loss (FCL2) and with spatial loos ($\mathcal{L}_{SL}$) is Fully Connected Neural Network with spatial loss (FCSL). In both cases, these models are trained by BP algorithm.

The BP algorithm is supervised learning because it uses labeled data to calculate the error. However, unsupervised learning allows models to self-learn the interpretable representation of data without the need for labeled data, such as deep generative models.

2.3.2 Deep Generative Modeling

Deep Generative modeling combines unsupervised representational learning and quantified uncertainty that provides probabilistic models, with the flexibility and scalability of deep neural networks. There are many deep generative models. We focus on GAN.

2.3.2.1 Generative Adversarial Network

Before explaining GAN, we present the main definitions related to this model. These are explained next.

Random Variables; a random variable is a variable that can take on different values at random. Therefore, it is just a description of the state that is possible. Moreover, a random variable must be coupled with a probability distribution that specifies how likely each of these states (Goodfellow et al., 2016).

A random variable may be discrete or continuous.

Probability Density Function; probability distribution depends on whether the variables are discrete or continuous and describe how likely random variables are to take on each of their possible states. When we have continuous variables, we can describe the probability distribution using Probability Density Function (PDF). To be a PDF, a function p must satisfy the following properties (Goodfellow et al., 2016):

- p must be the set of all possible states of x.
- $\forall x \in x, p(x) \geq 0$. Note that it does not require $p(x) <= 1$
- $\int p(x)dx = 1$

Conditional Probability; conditional probability is the probability of some event, given that some other event has happened. Therefore, the conditional probability that y=y given x=x is denoted as P(y=y | x=x), given as Equation 2.10 (Goodfellow et al., 2016).

$$\begin{aligned} P(\mathrm{y} = y|\mathrm{x} = x) &= \frac{P(\mathrm{y} = y, \mathrm{x} = x)}{P(\mathrm{x} = x)} \\ p(y|x) &= \frac{p(x, y)}{p(x)} \end{aligned} \tag{2.10}$$

Where the conditional probability is only defined if $p(x) > 0$.

Join probability; joint probability is the probability of an event occurring at the same time that another event occurs.

Bayes' Rules; in many situations we know $p(y|x)$, however, we need to know $p(x|y)$, then, if we know $P(x)$, we can calculate the desired quantity using Bayes' rule, which is defined as Equation 2.11 (Goodfellow et al., 2016).

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{2.11}$$

Discriminative models; discriminative models estimate the conditional probability of label y, given the features values $\bar{X}$. Therefore, the conditional probability is defined as $p(y|\bar{X})$ (Aggarwal, 2018).

Generative models generative models estimate the joint probability $p(\bar{x}, y)$, which is the generative probability of a data instance. Therefore, it can be used to estimate the conditional probability $p(y|\bar{X})$ by Bayes' rule, given as Equation 2.12.

$$p(y|\bar{X}) = \frac{p(\bar{X}, y)}{p(\bar{X})} \tag{2.12}$$

GAN (Goodfellow et al., 2014); GAN is a framework, which has two networks called G and D. G is a generative model and creates synthetic or fake data from input noise variables ($p_z(z)$) in order to learn a distribution p_g over real data with a distribution defined as $p_{data}(x)$. Then, fake and real data are candidates for D, D is a discriminative model and evaluates the validity of them, according to its output, which is a single scalar. While D is trained based on the distribution of real data to decrease its error, simultaneously G is trained to create synthetic data increasingly similar to real data and thus increasing the error of D.

Figure 2.5 illustrates a general scheme of the adversarial learning between G and D, because G tries to maximize the final classification error, while D tries to minimize the final classification error, respectively. In other words, these networks play the two-player minimax game, where the objective function is given as Equation 2.13 (Goodfellow et al., 2014).

$$\min_G \max_D \mathcal{L}_{cGAN}(G, D) = E_{x\ p_{data}(x)}[\log D(x)] + E_{z\ p_z(z)}[\log(1 - D(G(z)))] \tag{2.13}$$

Therefore, we have to train G and D to optimize Equation 2.13. Algorithm 1 shows a pseudocode for training a GAN model.

In the first iterations of GAN training, the D can identify the fake data generated by G and real data easily. It is because G creates only noisy data. Whether both networks have enough capacity in the time near convergence, the samples of G distribution

Algorithm 1 Algorithm for training the GAN

n: Number of learning epoch
$steps_G$: Number of steps to adjust G

$steps_D$: Number of steps to adjust D

Z^m: m samples from noise distribution $p_g(z)$

X^m: m samples from the data distribution
$p_{data}(x)$
m: number of sample mini-batch
for $i = 0 \rightarrow n$ **do**
    for $k_D = 0 \rightarrow steps_D$ **do**
        Get Z^m
        Get X^m
        Update the parameters of D:
        $\theta_D + \leftarrow - \nabla_{\theta D} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log (1 - D(G(z^{(i)})))]$
    end for
    for $k_G = 0 \rightarrow steps_G$ **do**
        Get Z^m
        Update the parameters of G:
        $\theta_G + \leftarrow - \nabla_{\theta G} \frac{1}{m} \sum_{i=1}^{m} \log (1 - D(G(z^{(i)})))$
    end for
end for

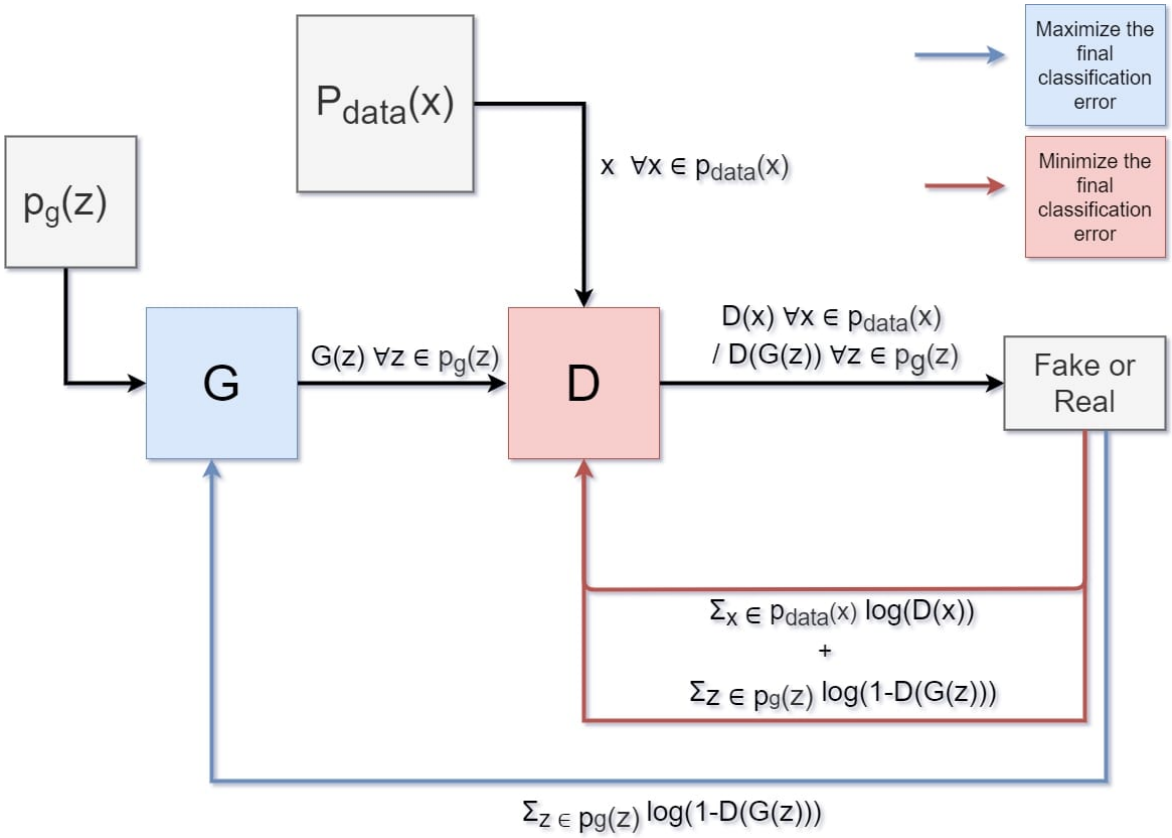


Figure 2.5: Generative Adversarial Network (GAN). source: Own.

will be similar to data distribution, then differentiating between these distributions will be difficult for D.

2.3.2.2 Conditional Generative Adversarial Network

cGAN is a variant of GAN. G and D come to be conditional when they are also affected for additional information y that is concatenated with z to take control of the synthetic data generation (Mirza and Osindero, 2014).

Thus, the objective function is given as Equation 2.14.

$$\min_G \max_D \mathcal{L}_{cGAN}(G, D) = E_{x\ p_{data}(x)}[\log D(x|y)] + E_{z\ p_z(z)}[\log(1 - D(G(z|y)))] \quad (2.14)$$

Therefore, adversarial loss functions for G and D are given as Equation 2.16 and Equation 2.15, respectively.

$$Loss_D = (log(D(x, y)) + log(1 - D(G(z|y), y)))/2 \quad (2.15)$$

$$Loss_G = log(D(G(z|y), y)) \quad (2.16)$$

2.3.3 Graph Attention Network

In a deep learning context, attention mechanisms are techniques that mimic cognitive attention, allowing one to focus on the most relevant factors to improve decision-making.

Attention mechanisms arose primarily as an improvement over the neural machine translation approach (Bahdanau et al., 2015). Before introducing this mechanism we explain some necessary concepts.

Recurrent Neural Network (RNN) Recurrent Neural Network (RNN) is a type of ANN that works well-processing sequences of vectors (Goodfellow et al., 2016). From the "traditional" ANN with one input vector and one output vector, there are types of RNN architectures, which are represented in Figure 2.6, where the model of "one to many" receives one vector and return sequences of output vectors. While "many to one" receives sequences of input vectors and the output is one vector. On the other hand, "many to many" receive a sequence of vectors and output is also a sequence of vectors, as well as, as this figure shows the architecture that is on the lower right the time of the last value of the sequence of inputs (x_{tx}) can be different to the time of the last value of sequences of outputs (y_{ty}).

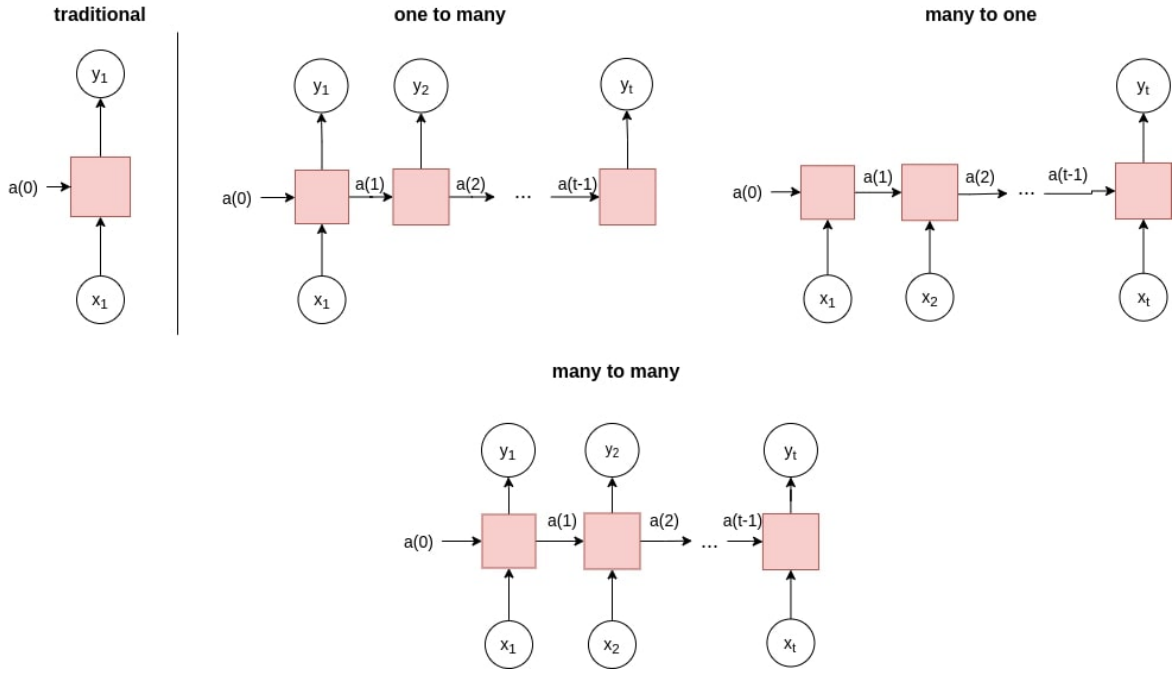


Figure 2.6: Recurrent Neural Networks.

Regardless of the architectures, in RNN previous outputs are used as inputs while having hidden states. Thus, in the forward step of this network for each timestep i the a_i and the output y_i are expressed as Equation 2.17 and Equation 2.18, respectively.

$$a_i = f(W_{aa}a_{i-1} + W_{ax}x_i + b_a) \tag{2.17}$$

$$y_i = g(W_{ya}a_i + b_y) \tag{2.18}$$

where W_{aa}, W_{ax} and W_{ya} are weight matrices respectively for hidden-to-hidden, input-to-hidden and hidden-to-output connections and b_a and b_y are bias vectors. Moreover, f and g are activation functions.

The loss function ($\mathcal{L}$) of all timesteps is defined by Equation 2.19

$$\mathcal{L}(\hat{Y}, Y) = \sum_{i=1}^{ty} \mathcal{L}(\hat{y}_i, y_i) \tag{2.19}$$

where $\hat{Y}$ is the sequence of output vectors of the RNN and Y is the sequence of desire vectors. In this case, for the update, the parameters of the network are using Backpropagation Through Time (BPTL), which is a variation of BP used on FCNN. Thus, BP is done at each point in time, then at timestep, t the derivative of the loss $\mathcal{L}$ with respect to a weight matrix W is defined as Equation 2.20:

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_{i=1}^{t} \frac{\partial \mathcal{L}_t}{\partial W}\bigg|_{(t)} \tag{2.20}$$

However, in the learning process of RNN the vanishing and exploding gradient phenomena often occur due to the difficulty to capture long-term dependencies because the temporal evolution of the backpropagated error can be exponentially decreasing or increasing with respect to the size of weights.

Therefore, Hochreiter and Schmidhuber (1997) proposed an architecture called Long Short Term Memory (LSTM) to avoid the vanishing and exploding gradient. LSTM can have many cells and each cell has three grates that regulate the flow of information into and out of the cell.

In the Figure 2.7, we can see cell of LSTM, where f denoted the forget gate, i denoted the input gate and o denoted the output gate. f is given as:

$$f = sigm(W_f[h_{t-1}, x_t] + b_f) \tag{2.21}$$

where the vector h_{t-1} denotes the output of a previous cell is concatenated with the vector x_t is the input vector to the current cell, then the resulting vector is obtained by applying the Sigmoidal function (sigm).

The input gate i is given as:

$$i = sigm(W_i[h_{t-1}, x_t] + b_i) \tag{2.22}$$

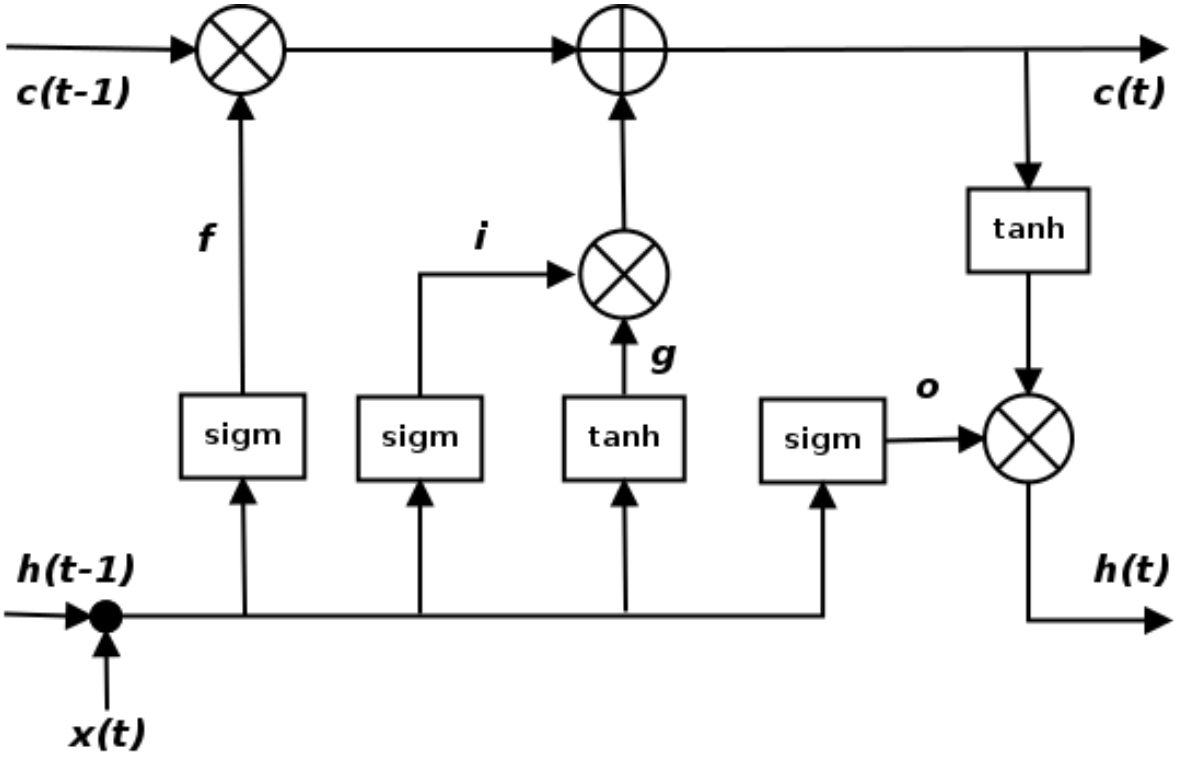


Figure 2.7: Long Short Term Memory

The output gate o is given as:

$$o = sigm(W_o[h_{t-1}, x_t] + b_o) \tag{2.23}$$

Moreover, C_t denoted the activation vector for the current cel, is given as:

$$C_t = f_t * C_{t-1} + i_t * tanh(W_C.[h_{t-1}, x_t] + b_c) \tag{2.24}$$

where, W denoted the weight matrix and b the bias vectors. While $tanh$ is the Hyperbolic Tangent Function.

Finally, h_t denoted the state of the certain cell in an instance t and is given as:

$$h_t = o_t * tanh(C_t) \tag{2.25}$$

The training of the whole architecture is usually using BPTL algorithm.

Encoder-Decoder The encoder-decoder architecture for RNN is a technique to address the sequence-to-sequence (Sec2Sec) proposed by and .

In the Encoder–Decoder framework, an encoder reads the input sentence, a sequence of vector $x = (x_1, , x_{T_x})$, into a vector c. Since a RNN approach we have Equation 2.26 and Equation 2.27.

$$h_t = f(x_t, h_{t-1}) \tag{2.26}$$

$$c = q(h_1, ..., h_{T_x}) \tag{2.27}$$

where $h_t \in \mathbb{R}^n$ is a hidden state at time t, and c is a vector generated from the sequence of the hidden states. f and q are some nonlinear functions.

The decoder defines a probability over the translation y by decomposing the joint probability into the ordered conditionals, given as Equation 2.28. Therefore, the decoder is often trained to predict the next word $y_{t'}$ given the context vector c and all the previously predicted words $\{y_1, ..., y_{t-1}\}$:

$$p(y) = \prod_{t=1}^{T} p(y_t | \{y_1, ..., y_{t-1}\}, c) \tag{2.28}$$

where $y = (y_1, ..., y_{T_y})$. With an RNN, each conditional probability is modeled as Equation 2.29.

$$p(y_t | y_1, ..., y_{t-1}, c) = g(y_{t-1}, s_t, c) \tag{2.29}$$

where g is a nonlinear, potentially multi-layered, function that outputs the probability of y_t, and s_t is the hidden state of the RNN.

Attention mechanism Bahdanau et al. (2015) proposed approach of jointly learning to align and translate achieves significantly improved translation performance over the basic encoder–decoder approaches.

Unlike the encoder-decoder approach defined as Equation 2.28, in the attention mechanism, the probability is conditioned on a distinct context vector c_i for each target word y_i, given as Equation 2.30.

$$p(y_i | y_1, ..., y_{i-1}, x) = g(y_{i-1}, s_i, c_i) \tag{2.30}$$

where s_i is an RNN hidden state for time i, computed by Equation 2.31.

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{2.31}$$

An encoder maps the input sentence to obtain a sequence of annotations $(h_1, ..., h_{T_x})$ in order to get the context vector c_i. In that sense, each annotation h_i represents information about the whole input sequence related to the $i-th$ word of the input sequence. Thus, the context vector is given as Equation 2.32

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2.32}$$

α_{ij} represents the weight of each annotation h_j, which is computed by Equation 2.33.

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{2.33}$$

where $e_{ij} = a(s_{i-1}, h_j)$, this scores how well the inputs around position j perform to match with position i.

The alignment model a is defined as a feed-forward neural network that is jointly trained with the whole proposed system. Basically, the α_{ij} is the probability that the target word y_i is aligned to, or translated from a source word x_j. Therefore, α_{ij} reflects the importance of the annotation h_j with respect to the previous hidden state s_{i-1} in deciding the next state s_i and generating y_i. Therefore, the decoder has an attention mechanism incorporated.

The attention mechanism is also currently used in other applications such as speech recognition, computer vision (Ramachandran et al., 2019; Chorowski et al., 2015), and node classification of structured graph data (Veličković et al., 2018), which is the application we focused on in this research.

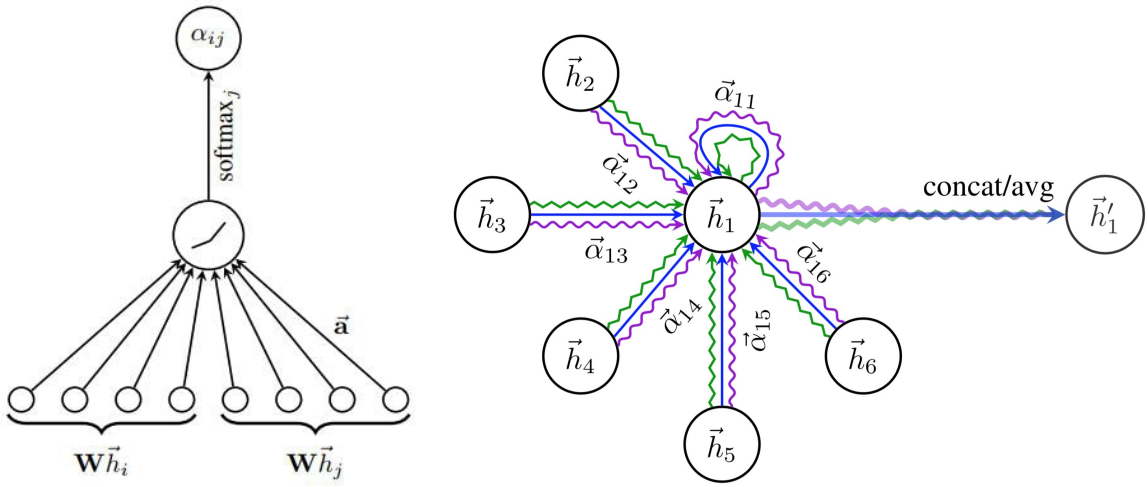
Veličković et al. (2018) proposed an attention-based architecture to make node classification of graph-structured data, called Graph Attention Network (GAT). GAT model computes the hidden representation of each node based on its similitude with neighbors according to the self-attention strategy.

GAT has a stacked graph attention layers. Each graph attention layer receives a set of node feature representations $h = \{\overrightarrow{h}_1, \overrightarrow{h}_2, \overrightarrow{h}_3, ..., \overrightarrow{h}_N\}$ $\overrightarrow{h}_i \in \mathbb{R}^F$ and it produces a transformed set of node feature representations h', $\overrightarrow{h}'_i \in \mathbb{R}^{F'}$, F' can be equals or different to F.

Figure 2.8a illustrates the attention mechanism. Feature representation of node i (h_i) from neighbor j (h_j) denotes the importance of node $j's$ features to node i based on a coefficient given as Equation 2.34

$$e_{ij} = a(W\overrightarrow{h}_i W\overrightarrow{h}_j) \tag{2.34}$$

The graph structure is injected into the mechanism by performing masked attention, computing e_{ij} for nodes $j \in N_i$, where N_i is some neighborhood of node i in the graph. To compare the coefficients calculated for each node j, the GAT's authors proposed to normalize the coefficients by Softmax function as Equation 2.35.



(a) (b)

Figure 2.8: (a) Attention mechanism given as e_{ij} (b) Multi-head attention with $K = 3$ by node 1 on its neighbors. Source: (Veličković et al., 2018)

$$\begin{aligned} \alpha_{ij} &= softmax(e_{ij}) \\ &= \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})} \end{aligned} \tag{2.35}$$

Moreover, they applied LeakyReLU function (Maas et al., 2013) to the calculated coefficients (Veličković et al., 2018) (Figure 2.8), which is given as Equation 2.36.

$$\alpha_{ij} = \frac{exp(LeakyReLU(\overrightarrow{a}^T[W\overrightarrow{h}_i || W\overrightarrow{h}_j]))}{\sum_{k \in N_i} exp(LeakyReLU(\overrightarrow{a}^T[W\overrightarrow{h}_i || W\overrightarrow{h}_k]))} \tag{2.36}$$

After obtain the coefficients for all neighbor nodes, they are used to compute the transformed feature representations h', according to Equation 2.37.

$$\overrightarrow{h}'_i = \sigma(\sum_{j \in N_i} \alpha_{ij} W \overrightarrow{h}_j) \tag{2.37}$$

where $\overrightarrow{h}'_i$ is the new transformed features for node i calculated based on the transformed features of its N_i neighbor nodes, $\overrightarrow{h}_j$ represent the features of neighbor node j and W is a weight matrix.

The GAT authors have found extending the mechanism to stabilize the learning process of self-attention, which is illustrated in Figure 2.8b. The K independent attention mechanisms execute the transformation of Equation 2.38.

$$\overrightarrow{h}'_i = ||_{k=1}^{K} \sigma(\sum_{j \in N_i} \alpha_{ij} W \overrightarrow{h}_j) \tag{2.38}$$

where $||$ represents concatenation, the $k-th$ attention mechanism (a^k) computes the α_{ij}^k, which are normalized attention coefficients and W^k is the corresponding input linear transformation's weight matrix. Then the output, h', will consist of KF' features.

Finally, the authors explained that preform multi-head attention on the final prediction layer of the network, concatenation is no longer sensible, instant, they apply to average and delay applying the final nonlinearity, which can be Softmax or Sigmoid function. This is given as Equation 2.39.

$$\overrightarrow{h}'_i = \sigma(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N_i} \alpha_{ij}^k W^k \overrightarrow{h}_j) \tag{2.39}$$

k-Nearest Neighbor attention pooling layer; Ma and Zhang (2019) propose the Affinity Network (AffinityNet), this network consists of stacked kNN attention polling layers. This layer is a generalization of GAT and it was proposed under the consideration that nodes belonging to the same cluster should have similar representations that are near the cluster center in clustering/classification tasks. Mainly, the difference between GAT and AffinityNet, is that the first model was proposed to address representation learning on knowledge graphs, while AffinityNet was proposed to facility representation learning on any collections of nodes with or without known graph.

Figure 2.9 shows the AffinityNet, where the input layer is followed by a feature attention layer, and then followed by multiple stacked attention layers, where the output of the last layer will be the newly learned network representations, which can be used for classification or regression tasks (Ma and Zhang, 2019).

kNN attention polling layer incorporates the neighborhood information using attention-based polling, given as Equation 2.40.

$$h'_i = f(\sum_{j \in N(i)} a(h_i, h_j) \cdot h_j) \tag{2.40}$$

where h_i and h'_i are input feature representation and transformed feature representation for node i, respectively. The neighborhood of i is represented by $N(i)$, where h_j represents the feature representation of node h_j. Then $a(h_i, h_j)$ represents the normalized attention from node i to node j and $a(.,.)$ is the attention kernel selected, which could be the following kernels (Ma and Zhang, 2019):

- Cosine similarity

$$\alpha_{ij} = \frac{h_i \cdot h_j}{||h_i|| \cdot ||h_j||} \tag{2.41}$$

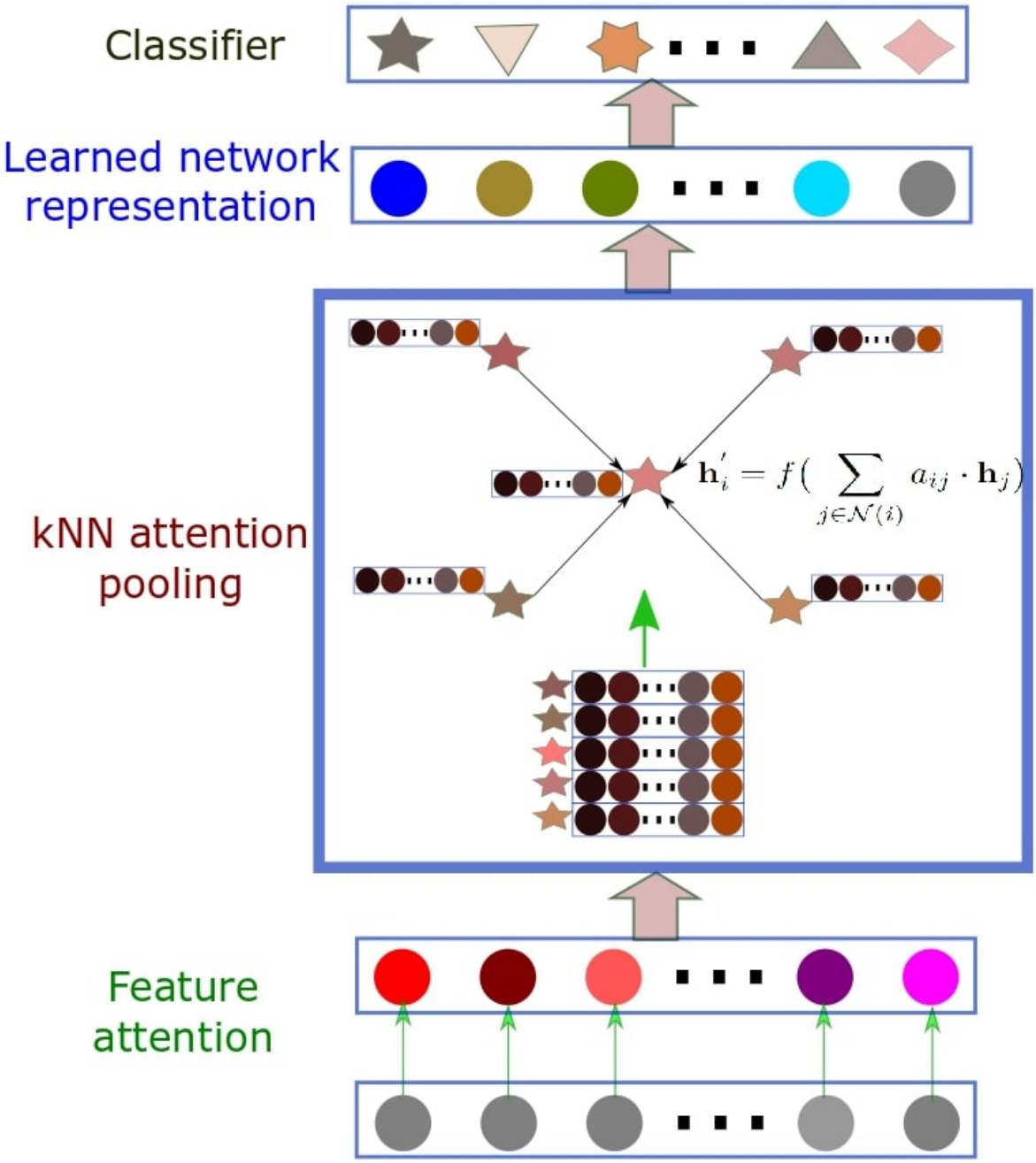


Figure 2.9: Affinity Network Model. source: (Ma and Zhang, 2019).

- Inner product

$$\alpha_{ij} = h_i \cdot h_j \tag{2.42}$$

- Perceptron affine kernel (based on (Veličković et al., 2018))

$$\alpha_{ij} = w^T \cdot (h_i || h_j) \tag{2.43}$$

- Inverse distance with weighted L_2 norm

$$\alpha_{ij} = -||w \odot h_i - w \odot h_j||^2 \tag{2.44}$$

The attention kernels calculate the similarities among nodes to facilitate weighted polling. The normalization of the output of the attention kernel to calculate a weighted average of feature representations is based on the Softmax function, given as Equation 2.45.

$$a_{ij} = a(h_i, h_j) = \frac{e^{\alpha_{ij}}}{\sum_{j \in N(i)} e^{\alpha_{ij}}} \tag{2.45}$$

2.4 Land-related variables

We considered land-related variables that are known to be associated with air pollution, where areas of high vegetation can have less pollution than barren areas (Wei et al., 2019; Wang et al., 2019). Also, the morphology and elevation of the terrain can influence the estimation of $PM_{2.5}$ concentration levels (Ma et al., 2017).

2.4.1 Normalized Difference Vegetation Index

NDVI measures the land vegetation levels and is obtained from satellite images. It is calculated as the difference in the Earth-Resources Technology Satellite (ERTS) radiance value measured in bands 5 and 7, divided by their sum defined as Equation 2.46.

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{2.46}$$

where NIR is the reflectance in the near-infrared region of the spectrum (0.7 to 1.1 μm) and Red is the reflectance at red wavelengths (0.6 to μm). NDVI is a value between -1 and 1, where values equal to or less than 0.1 indicate empty and barren areas or snow, while values between 0.2 to 0.3 represent shrubs and meadows and longer values represent tropical forest areas. We obtained NDVI data from MOD13A2 product (Didan, 2015) with a resolution of 1km per pixel.

Figure 2.10 shows the index of NDVI on a region of Saõ Paulo, where the minimum value is -0.8 and areas with high vegetation have values close to 0.7.

2.4.2 Digital Elevation Model

Digital Elevation Model (DEM) is a land surface model and a raster representation of a continuous surface, which does not include trees, buildings, or other non-ground objects. We obtained DEM data with a high resolution of 30m per pixel and from a public repository offered by the mission Shuttle Radar Topography Mission (SRTM) launched by the National Geospatial-Intelligence Agency (NASA) and the German and Italian Space Agencies in February 2000 (Farr et al., 2007). Figure 2.11 shows a DEM in a region of São Paulo obtained from Shuttle Radar Topography Mission (SRTM).

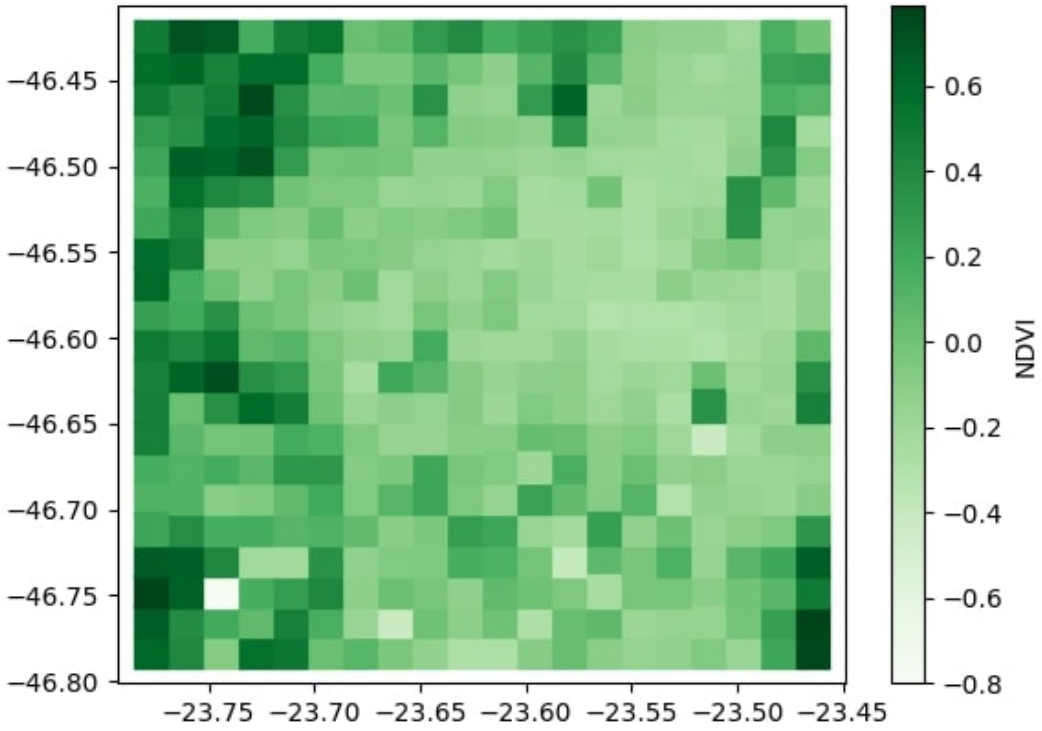


Figure 2.10: Index of NDVI on a region of Saõ Paulo.

2.5 Population-related variable

2.5.1 Nighttime Lights

Another source of information that may be associated with air pollution is the NTL index. This variable is important because regions with a high NTL index indicate areas of high population density that may be associated with high emissions of pollutants. NTL measures the brightness and the spatial extent of light intensity on the surface from the Day/Night Band (DNB) detectors, which are part of the Suomi-NPP Visible Infrared Imaging Radiometer Suite (VIIRS). VIIRS monitors the intensity of nighttime and anthropogenic sources of light emissions at a resolution of 500m per pixel. Therefore, NTL allows obtaining knowledge about human and economic activities or population levels in a region. We obtained NTL from the VNP46A1 product (DAAC). Figure 2.12 shows the NTL over the earth's surface.

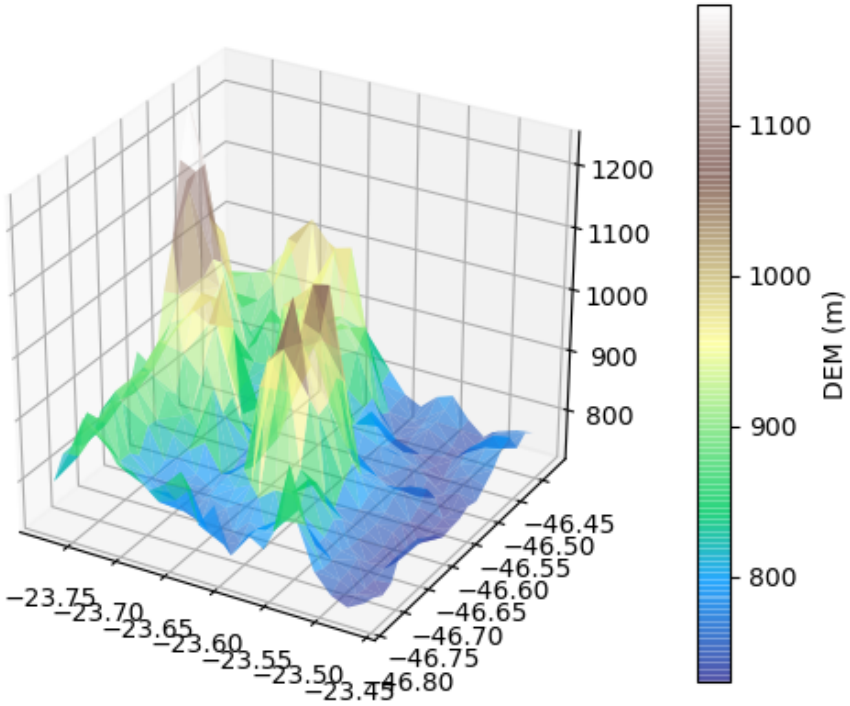


Figure 2.11: Digital Elevation Model (DEM) of a São Paulo region obtained from Shuttle Radar Topography Mission (SRTM).

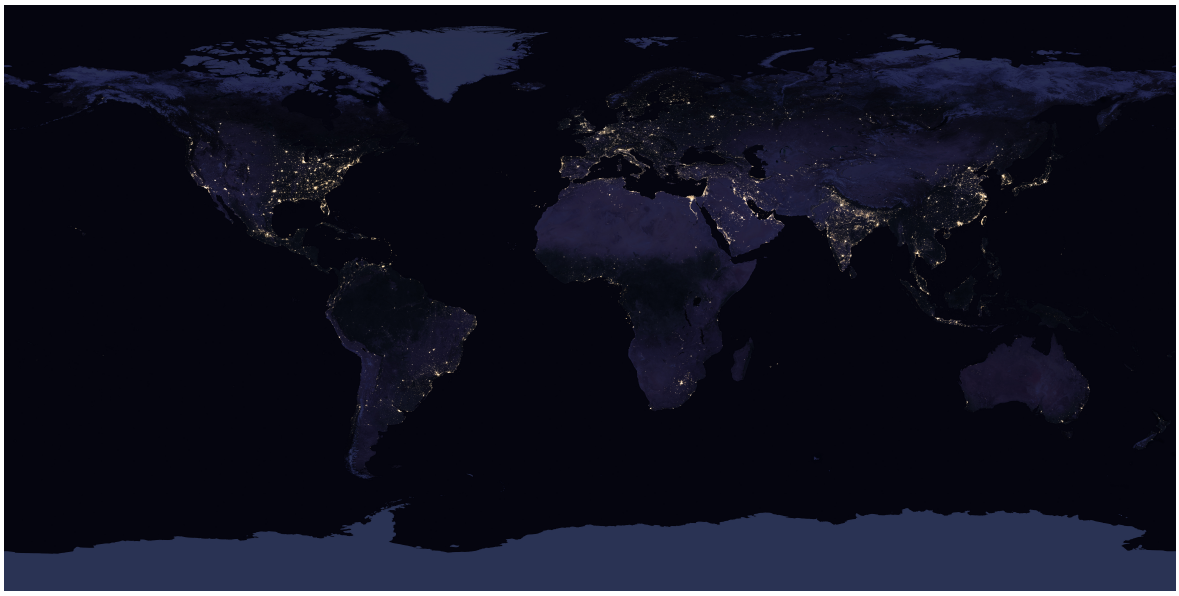


Figure 2.12: NTL over the earth's surface using the NASA VIIRS DNB algorithm. source: (Kalb et al., nd)

Chapter 3

Related Work

There are two methods to predict and simulate the diffusion and dispersion of air pollutant concentrations: the Deterministic methods and the Statistic methods.

3.1 Deterministic Methods

The deterministic methods adapt meteorological principles based on atmospheric physics and chemical models to estimate and simulate the diffusion and dispersion of pollutant concentrations. There are two main models: Weather Research and Forecasting model coupled with Chemistry (WRF-Chem) and Community Multiscale Air Quality Modeling System (CMAQ). WRF-Chem is used to estimate air pollutant concentrations. Also, this model is used to assess the relationship between air pollutants and other factors such as; geographic factors, traffic flow, and weather conditions (Reátegui-Romero et al., 2018; Sánchez-Ccoyllo et al., 2018; Saide et al., 2011).

On the other hand, Chen et al. (2014) used the CMAQ Version 4 model (Foley et al., 2010) to simulate $PM_{2.5}$ formation for regulatory applications perform the control of emission.

However, these methods are complex due to the number of variables that affect the diffusion and dispersion of air pollutants. Thus, the necessary data collection is difficult to perform and experiment with these types of models (Delavar et al., 2019).

3.2 Statistical Methods

The statistical methods only apply modeling statistics to predict air pollutant concentrations. The research studies focused on the temporal prediction of air pollutant concentrations have used traditional statistics models, such as Multiple Linear Regres-

sion (MLR) (Paschalidou et al., 2011) and Autoregressive Integrated Moving Average (ARIMA) (Wang and Guo, 2009; Zhang et al., 2018b). Support Vector Regression (SVR) is another model widely used for the prediction of air pollutants. Nieto et al. (2013) perform air pollution modeling applying SVR to get the better hyperparameter, then they estimated the dependence between primary and secondary air pollutants. In other research, Osowski and Garanty (2007) perform the prediction of four pollution factors CO, NO_2, SO_2 and dust using the Support Vector Machine (SVM) and six types of wavelet transformation applied to time series obtained of seven monitoring stations in the northern region of Poland. In addition, they have shown the comparison results between SVM and Multilayer Perceptron (MLP), where SVM outperforms the MLP.

There are many variables related to air pollution that reduce the performance of traditional shallow statistical models. Thus, some research studies proposed the ANN for the prediction of air pollutant concentrations, such as MLP and Radial Basis Function Neural Network (RBFNN) (Zou et al., 2015). On the other hand, Deep learning models provide a good representation of the complicated process of air pollution (Li et al., 2016; Pruthi and Liu, 2022).

In this sense, Li et al. (2016) proposed a novel model called Spatio-temporal Deep Learning, which is the concatenation of a Stacked Autoencoder (SAE) and Logistic Regression (LR). SAE extracts the most relevant spatio-temporal features of air quality based on unsupervised learning, while LR fits the $PM_{2.5}$ concentration prediction of the whole model. Moreover, they compared this model with Spatiotemporal Artificial Neural Network (STANN) (Anh Nguyen et al., 2012), Autoregressive Moving Average (ARMA) and SVR, and their model was effective and outperformed the other models.

Moreover, the air quality and meteorological variables are temporal sequences. Thus, the RNN can represent their temporal behavior. Li et al. (2017b) proposed an extension of the LSTM, which is called Long Short Term Memory Extension (LSTME), this model has a LSTM to process the time series of $PM_{2.5}$ concentrations, which were recollected by 12 monitoring stations in Beijing. This result is joined with auxiliary data (normalized meteorological and timestamp information), then this is processed by an additional fully connected layer to get the final prediction of $PM_{2.5}$ per each AQM station. In addition, the authors performed the comparison of their model with a Long Short-Term Memory Neural Network (LSTM NN), Spatiotemporal Deep Learning (STDL), SVR, ARMA, and Time Delay Neural Network (TDNN). They state that their model outperformed the others.

However, in a deep learning context, the attention mechanism has performed better in machine translation than RNN (Bahdanau et al., 2015). Therefore, models based on that mechanism have been proposed to predict air pollutant concentrations using additional factors related to the pollution process and capturing Spatio-temporal dependencies of air pollutant concentrations at the same time (Li et al., 2020; Shi et al., 2021). Recently, models based on GAT have been developed to forecast air quality because they can obtain information about dynamic iteration between the neighbors to form a neighborhood graph where nodes are weighted according to the differences

between them (Iskandaryan et al., 2023; Li et al., 2021). Li et al. (2021) proposed a model with GAT to predict $PM_{2.5}$ concentrations, where the dynamic graph allowed the identification of neighbors' information for each node in each forecasting period and determined the link weight values. That model achieved better results than the baseline models to predict air quality.

The mentioned models so far only make predictions for future times at observed points because it is difficult to obtain values representing the air pollution conditions over a full research region where the AQM stations are uneven and sparsely distributed (Zhang et al., 2018a). In this regard, studies have focused on the estimation of air pollutant concentrations in locations without AQM stations by traditional interpolation methods such as; IDW and OK (Wong et al., 2004; Li et al., 2014; Masroor et al., 2020; Vargas-Campos and Villanueva, 2021). Nevertheless, shallow-interpolation models generally have unsatisfactory results because they may not be able to extract information on temporal dependencies, the emission, and the dispersion of air pollutant concentrations due to the several factors that affect the pollution process (Zheng et al., 2013; Wei et al., 2019; Wang et al., 2019).

Therefore, Deep Learning models have been proposed for the spatial prediction of air pollutants. Guo et al. (2020); Ma et al. (2019); Fan et al. (2017); Pruthi and Liu (2022) proposed models of deep learning combined with IDW for Spatio-temporal prediction of air pollutant concentrations. Even better, Qi et al. (2018) proposed a novel model called Deep Air Learning (DAL). They used data from 12 AQM stations in Beijing, which is data labeled. However, the uneven monitoring stations produce unlabeled data in many locations. Thus, they used an autoencoder adjusted by Spatio-temporal Semisupervised Learning to solve these limitations, based on the spatial and temporal kNN information. Figure 3.1 shows the model proposed in that research. In that figure, there are three components. The first component is an autoencoder with a sparse layer to the selection of relevant and irrelevant factors related to $PM_{2.5}$ and the last component calculates the spatial and temporal loss based on the observed values of the near AQM stations and values previously observed for a certain instance, respectively.

Other studies proposed deep learning models and utilized Aerosol Optical Depth (AOD) as the main-input feature to predict air pollutant concentrations. AOD is a measure of light scattering and absorption for particle concentrations in a vertical column of the atmosphere, which have a relationship with $PM_{2.5}$ concentrations based on some studies (Shin et al., 2020; Yang et al., 2019). Additionally, auxiliary information obtained from other satellite products, as well as NDVI and NTL, which indicate the level of vegetation of the surface and human activity, respectively (Li et al., 2017a; Di et al., 2019). These studies achieved predicted air quality accurately and found the relationship between the air pollution process, NDVI and NTL. However, due to uncertainties such as cloud screening schemes and assumptions in retrieval algorithms (Yang et al., 2019; Sayer et al., 2013), there are limitations in using AOD as a key feature to predict $PM_{2.5}$.

In a deep learning context, the generative models learn the probability distribu-

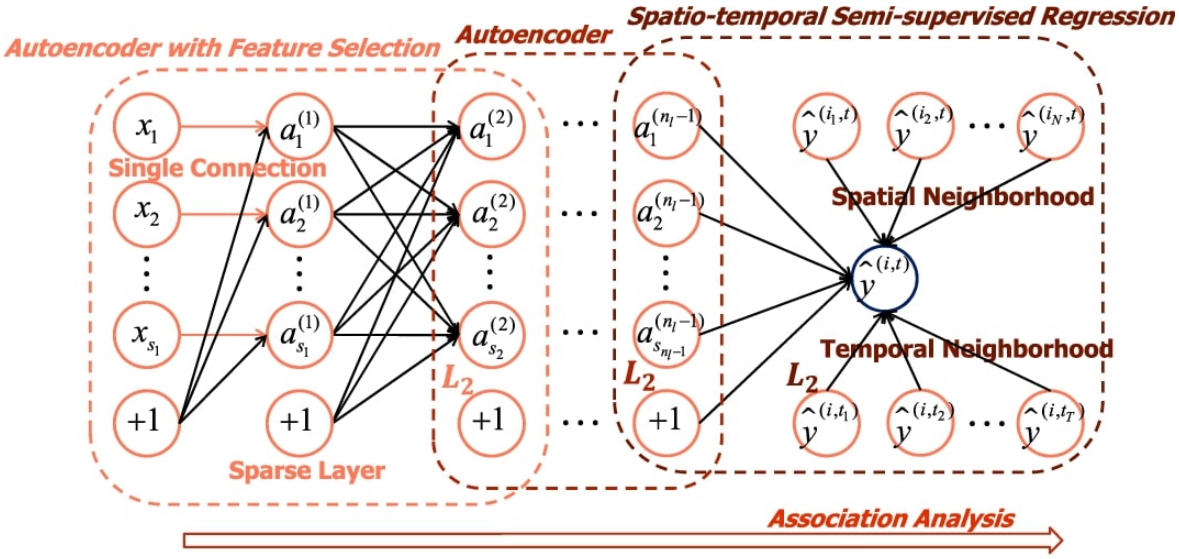


Figure 3.1: Deep Learning model to interpolate and predict fine particulate matter

tion of the real-world data to generate synthetic data similar to real data. A well-known model generative model is GAN, this model captures the patterns of the real data based on adversarial learning to generate synthetic data. GAN and variants of it have been good results in Computer Vision and Artificial Intelligence fields Gui et al. (2020); Alqahtani et al. (2021); Jabbar et al. (2021). In this sense, GAN has been proposed in the air pollution context. For example, Toutouh (2021) proposes models based on cGAN to predict $PM_{2.5}$ concentrations and Friedjungová et al. (2020) proposes a model based on GAN called Wasserstein Generative Adversarial Imputation Network to impute synthetic data on unobserved points. Gao et al. (2020) design a novel model called Spatial Interpolation with Attentional Generative Adversarial Networks, which includes a S^2 attention structure embedded in GAN. The most relevant thing about this article is that without any additional information. Therefore, this model can learn the correspondence between low/high-quality signals. In addition, the attention-based structure S^2 extracts global features allow avoiding increasing the size of convolution layers and convolution kernels.

In this research, we propose as a first method a model based on generative modeling, known as cGANSL, which is a variation of GAN. The conditional data are meteorological and kNN information for spatial prediction of $PM_{2.5}$ concentrations. Moreover, we add spatial learning. Nevertheless, the most statistical methods, deep learning models, including the proposed cGANSL need the identification of the kNN stations based on straight-line distance, usually Euclidean distance, for air quality spatial prediction, then the selection of suitable k is based on the calculated spatial distances. However, the nearby information often is conditioned by many factors such as meteorological variables, variables related to earth and related to human activity.

Therefore, we proposed a second model, which is a neural network with an attention-based layer called kNN attention polling layer, which is a component of

AffinityNet (Ma and Zhang, 2019). AffinityNet is a model based on GAT and it was proposed to predict disease types. Moreover, the authors of AffinityNet state that an attention-based layer can calculate important features that are useful for classification tasks. In this sense, we use kNN attention polling layer to calculate the attention levels from each node of a structured graph to its kNN nodes by an attention kernel. Each node in the graph has variables related to earth and population represented by NDVI and NTL, respectively, as well as meteorological variables and $PM_{2.5}$ concentrations observed. Then attention-based layer creates transformed feature representations for each node, which are entered into FCNN to predict $PM_{2.5}$ concentrations.

Chapter 4

Preprocessing and matching data

This chapter shows the process to prepare and match the data for the training and testing steps of the proposed spatial prediction models. We use data on meteorological conditions, such as Temperature (T), Pressure (P), Relative Humidity (RH), Dew Point (DP), Wind Direction (WD) and Wind Speed (WS), as well as, variables related to land (NDVI and DEM) and population (NTL). Moreover, we use data of level of $PM_{2.5}$ concentrations collected by AQM stations. In order to obtain the pre-proposed data, we perform the following steps.

4.1 Preprocessing data

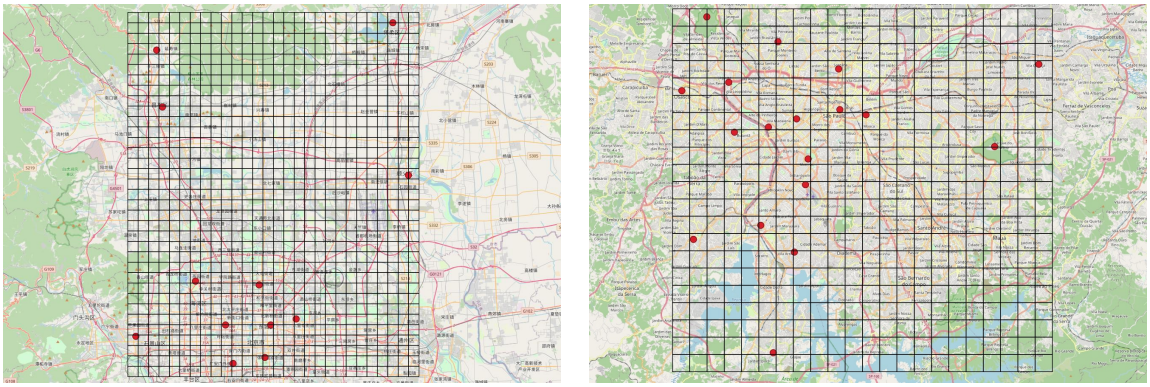
4.1.1 Cleaning missing values

Commonly, the information on air pollution and meteorological variables are ground-level data, which are recorded by AQM stations and meteorological stations. This information is hourly and can have some intervals with missing data due to the malfunction of sensors implemented in these stations. Therefore, we first clean and analyze this data to improve the learning performance of the proposed spatial-prediction models in this research.

4.1.2 Map division

We defined a bounding box that covers all meteorological and AQM stations using the QGIS software. Figure 4.1a and Figure 4.1b show the coordinate bounding box defined on the map of Beijing and São Paulo used in this research, respectively. As well as the red points represent the AQM stations and meteorological stations.

We create cells of 2km of resolution in the coordinate bounding box. Some cells



(a) (b)

Figure 4.1: (a) Coordinate bounding box defined on Beijing map. (b) Coordinate bounding box defined on São Paulo map.

may have an AQM station in their areas. Thus, Cells with a located AQM station are defined as labeled points. Otherwise, they are unobserved points.

4.1.3 Cropping information from satellite imagery

We use information from satellite images, such as NTL and NDVI, using the Level-1 and Atmosphere Archive and Distribution System Distributed Active Archive Center (LAADS DAAC) website to obtain them. To complete the request on the website, we input the interval dates determined from the ground-level data (air quality and meteorological data) and the previously defined coordinate bounding box. Then, LAADS DAAC sent us an email with the generated endpoints to obtain the data.

The data obtained are images where the coordinates are matched based on the region of the bounding box. Thus, we realized a limitation in this step concerning the satellite data. First, in the best case, a collected image can cover all the cells of the coordinate bounding box. For example, Figure 4.2 shows the satellite images requested based on the coordinate bounding box defined on the AQM network of São Paulo and the cropping process performed to obtain only information within the coordinate bounding box. However, in other cases was necessary to collect two images to cover the bounding box. For example, in the case of the bounding box defined on the AQM network of Beijing, we had to obtain two images to cover the whole region.

On the other hand, DEM was obtained from the website of United States Geological Survey (USGS). In this step, we entered the coordinate bounding box to the website gave us one o more images with DEM information depending on the matching with the bounding box. Therefore, we had to pre-process this information to obtain only the data within the defined bounding box.

NDVI has a daily frequency of 16 days, NTL has a daily frequency. For normalization of all data, NDVI data was also up-sampled to daily frequency filling the

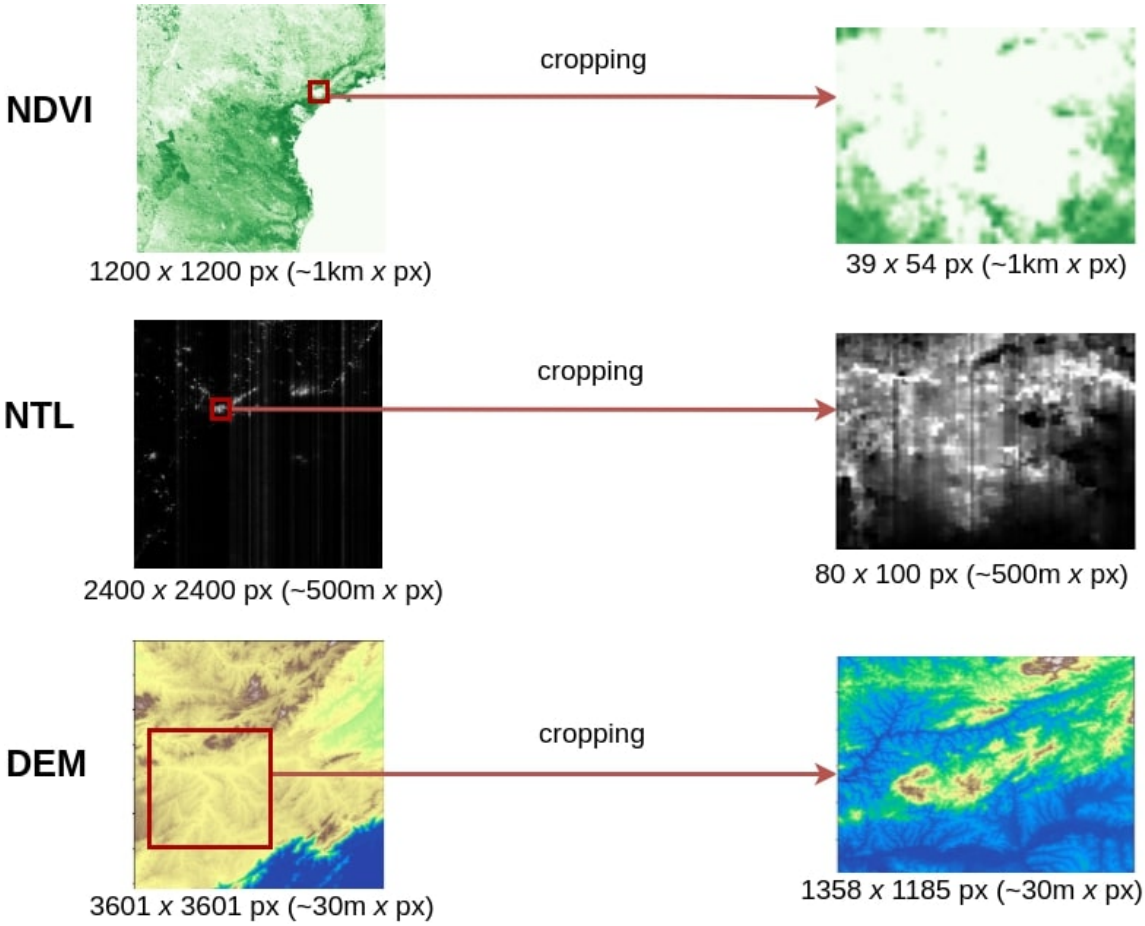


Figure 4.2: Matching data. source: Own.

closest past observation for each daily record. DEM is independent of time. In this sense, we also aggregated $PM_{2.5}$ and meteorological records to daily frequency based on the mean hourly.

4.2 Matching data

Figure 4.3 shows the process of matching all variables. We get the weather conditions (TEMP, RH, PRESS, WD) from the nearest meteorological monitoring station for a location and day of a determined period. Since NDVI has a resolution of 1km per pixel, we resize it to 2km based on the mean of NDVI values to obtain a value per grid of the bounding box. NTL have a resolution of 500m. Therefore, we compute the mean of four values to resize it to 2km. Similarly, we calculated the mean of DEM values because it has a resolution of 30m per pixel.

Moreover, if there is a AQM station at a grid, we obtain the concentration of $PM_{2.5}$ observed at that location and this is classified as labeled. Otherwise, it is unlabeled.

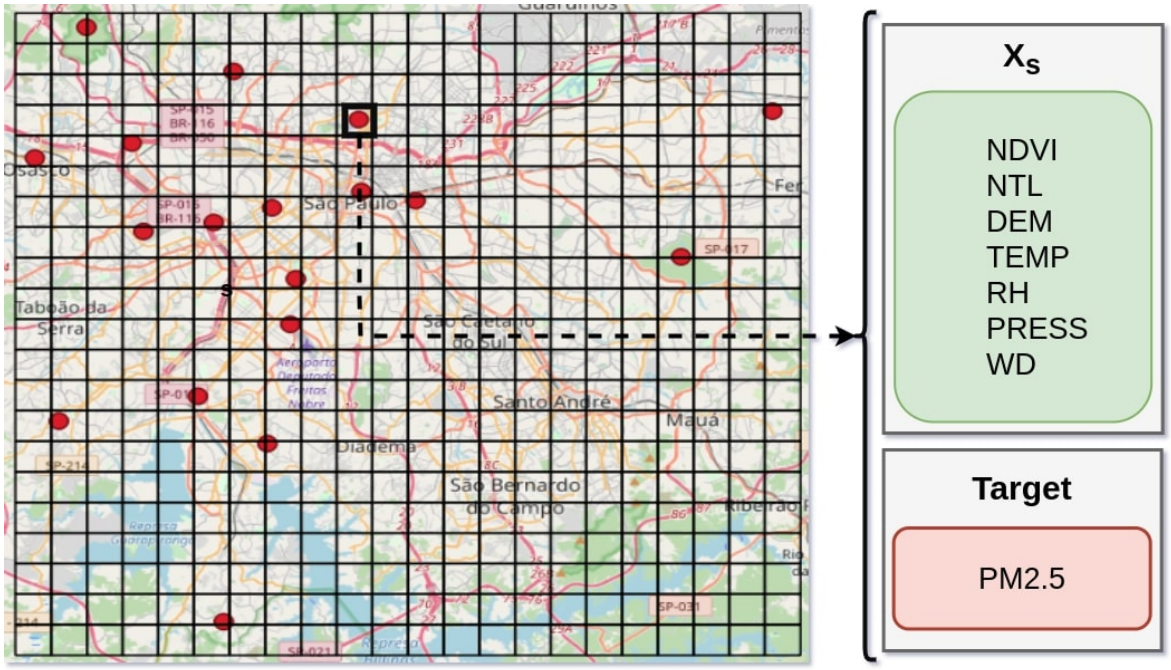


Figure 4.3: Matching data. source: Own.

Chapter 5

Method I: cGAN and Spatial Learning for Spatial Prediction of Fine Particulate Matter

We propose a cGAN model to predict $PM_{2.5}$ concentrations in unobserved locations. cGAN is an extension of GAN, which has two ANN: (1) G network can generate synthetic data and (2) D network evaluate the authenticity of the data. cGAN is explained in more detail in Section 2.3.2.2. Moreover, we adjusted the cGAN for spatial prediction of $PM_{2.5}$ concentrations, considering two steps: training and testing step explained below.

5.1 Training step

Figure 5.1 shows the training step of cGAN, for each instance $\{t_1, t_2, ...\}$, we selected labeled data (locations with an AQM station) and we remove location where is the station selected for testing phase. After for each location y_* G creates Synthetic Fine Particulate Matter ($PM_{2.5}*$) from the input data, which is a joined vector between z and the conditional information (y), such as meteorological variables, variables related to the earth and population (NDVI, DEM and NTL), as well as, the $PM_{2.5}$ concentrations observed by the kNN stations. Then $PM_{2.5}*$ is joined with y vector to form the synthetic data. On the other hand, the real data are the union of the concentrations $PM_{2.5}$ observed by the stations and the vector y.

At the same time, the D tries to estimate the authenticity of the synthetic data and the real data correctly. Thus, the D is trained based on its error when it attempts to distinguish the authenticity correctly (the blue dotted line in Figure 5.1). While the G is trained according to its error when it attempts to lie to the D with its generated synthetic data (the red dotted line in Figure 5.1).

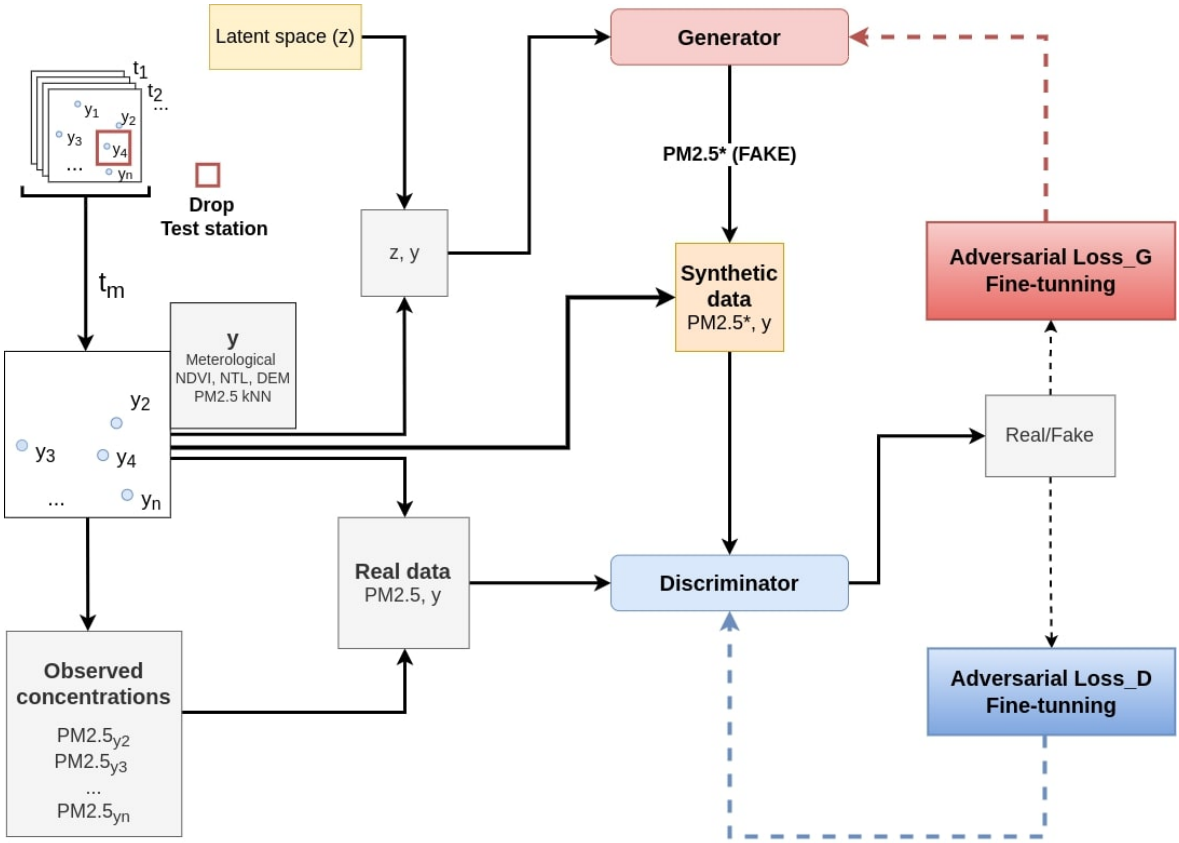


Figure 5.1: Training of cGAN for spatial prediction of $PM_{2.5}$, source: Own.

Therefore, adversarial loss function for G and D are given as Equation 5.1 and 5.2, respectively:

$$Loss_G = log(D(PM2.5*, y)) \tag{5.1}$$

$$Loss_D = (log(D(PM2.5, y)) + log(1 - D(PM2.5*, y)))/2 \tag{5.2}$$

5.2 Adversarial Learning and Spatial Learning

Since the close observations in space and time tend to be more similar than those far away (Li et al., 2016; Cressie and Wikle, 2011), we added a loss calculation based on a spatial approach for labeled and unlabeled points. This loss is calculated for each predicted $PM_{2.5}*$ of an instance i based on the weighted difference with the values observed by its kNN stations.

Thus, we calculate weights for each instance based on the Softmax function of the spatial distances to its kNN, given as Equation 5.3.

$$w^s = softmax(d^{-1}) \tag{5.3}$$

where d is the vector of spatial distances between the location of the instance i to its kNN stations. Therefore, the spatial loss is given as Equation 5.4.

$$SLoss = \sum_{k=1}^{K} |PM2.5 *_i - PM2.5_k| * ws_{ik} \ \forall i \in N \tag{5.4}$$

where:

- $PM_{2.5}*_i$ is $PM_{2.5}*$ predicted for the instance i by G.
- $PM2.5_k$ is $PM_{2.5}$ observed by the k nearby AQM station.
- ws_{ik} is weight calculated based on the spatial distance between the location of the instance i and the k nearby AQM station.

The smoothing applied in this approach adds to the model the ability to make predictions according to spatial distance. Spatial Loss (SLoss) is included in the G adversarial loss represents as the green dotted line in Figure 5.2.

In addition, the spatial loss is calculated for all the grids of the bounding box due to labeled and unlabeled points having kNN observations of $PM_{2.5}$. In contrast, the adversarial learning of cGAN only applies to labeled locations. Therefore, for each learning epoch, we select labeled data to calculate the loss function by Equation 5.1. While for labeled and unlabeled data, SLoss is calculated using the equation Equation 2.9. Therefore, the new loss function is calculated by Equation 5.5. This extension we called cGANSL.

$$Loss_{G_{SL}} = (\lambda * \frac{1}{n}(\sum_{i=0}^{n} Loss_{G_i})) + (\sigma * \frac{1}{m}\sum_{j=0}^{m} SLoss_j) \tag{5.5}$$

where $Loss_{G_i}$ is G adversarial loss for instance i,λ is the adversarial parameter, σ is the spatial parameter, n is the number of labeled points for training epoch and m is the number of labeled and unlabeled points for training epoch.

5.3 Testing step

For each training step, we evaluate the trained model using labeled data from a selected testing AQM station (not included in the training step). Figure 5.3 shows the generation of synthetic concentrations of $PM_{2.5}$ through of trained G after the training step. Then the generated values are compared with the $PM_{2.5}$ concentrations observed in the testing station to obtain the spatial prediction performance of the model.

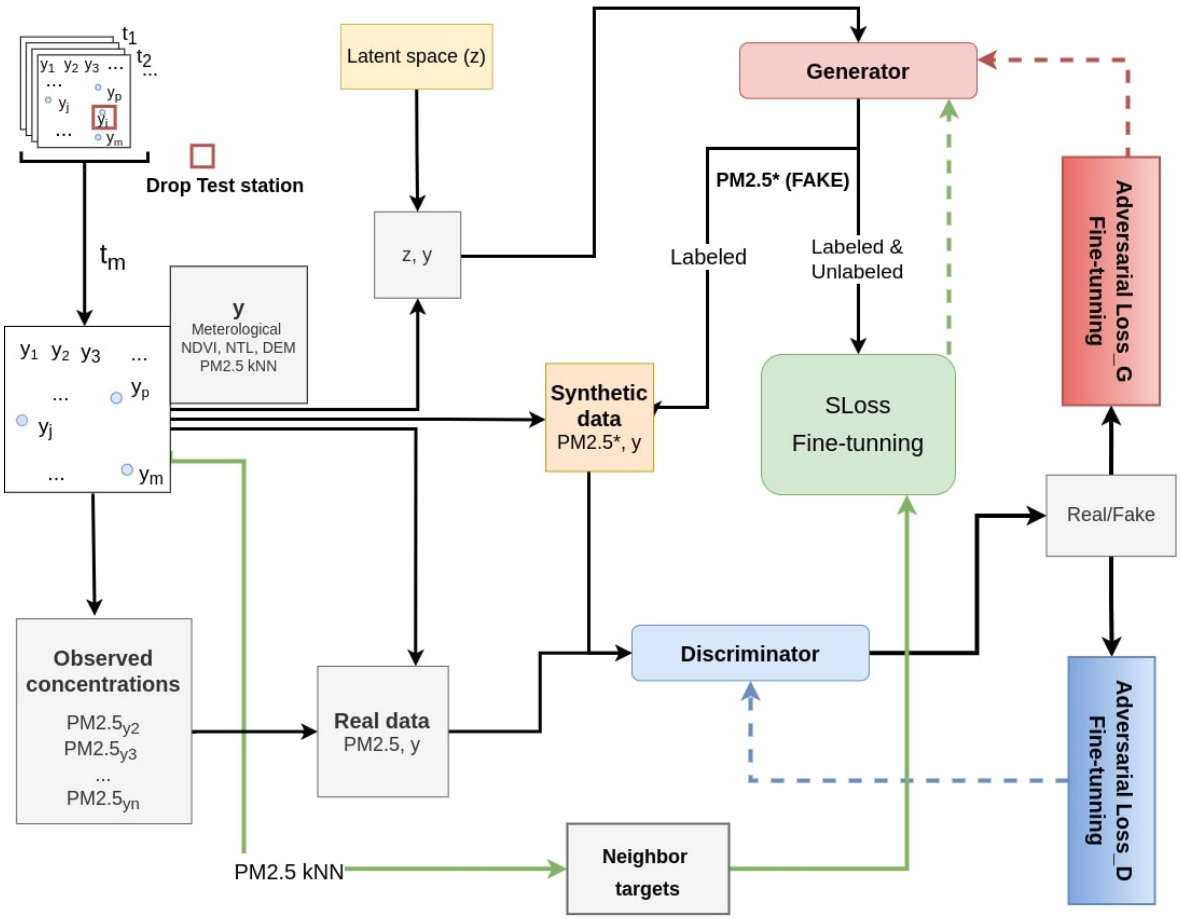


Figure 5.2: Training of cGANSL for spatial prediction of $PM_{2.5}$, source: Own.

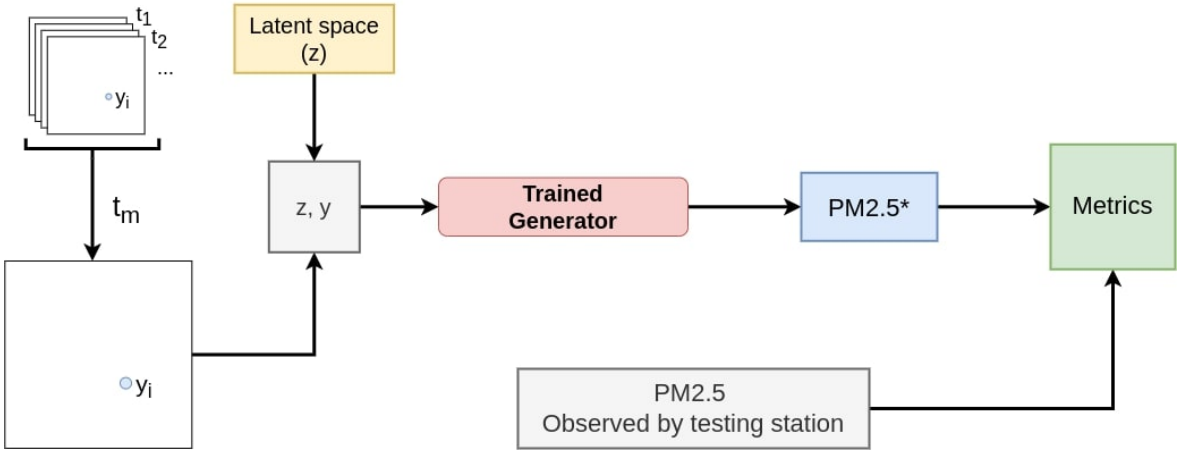


Figure 5.3: Testing of cGAN and cGANSL for spatial prediction of $PM_{2.5}$, source: Own.

Chapter 6

Method II: Neural Network with Attention-based Layer for Spatial Prediction of Fine particulate matter

We propose kNN attention polling layer (see Section 2.3.3) and FCNN to predict $PM_{2.5}$ concentrations spatially. Since each location with certain data can be represented by a node in a graph, the attention layer allows calculating a transformed feature representation for each node based on the attention given to its kNN nodes, which are nodes where are available an observed $PM_{2.5}$ concentration. Then these representations are entered to FCNN (see Section 2.3.1.1) to obtain the predicted $PM_{2.5}$ concentration for each node.

6.1 KNN attention polling layer

The kNN attention polling layer (see Section 2.3.3) is proposed to incorporate neighborhood information using an attention mechanism, which includes an attention kernel to calculate the attention given from a node to its kNN nodes. The attention kernel can be Cosine Similarity (Equation 6.1), Inner Product (Equation 6.2), Perceptron Affine (Equation 6.3) or Inverse Distance (Equation 6.4).

- Cosine Similarity

$$\alpha_{ij} = \frac{X_i \cdot X_j}{||X_i|| \cdot ||X_j||} \tag{6.1}$$

- Inner Product

$$\alpha_{ij} = X_i \cdot X_j \tag{6.2}$$

- Perceptron Affine kernel (based on (Veličković et al., 2018))

$$\alpha_{ij} = w^T \cdot (X_i || X_j) \tag{6.3}$$

- Inverse Distance with weighted L_2 norm

$$\alpha_{ij} = -||w \odot X_i - w \odot X_j||^2 \tag{6.4}$$

Note that we use X data instead of transformed feature h as is given in equations in Section 2.3.3 because we connected the input features directly to the attention process.

Therefore, the attention kernels calculate the level of attention based on the meteorological conditions and the land-related variables of a node to its kNN nodes denoted as a_{ij} and given as Equation 6.5.

$$a_{ij} = a(h_i, h_j) = \frac{e^{\alpha_{ij}}}{\sum_{j \in N(i)} e^{\alpha_{ij}}} \tag{6.5}$$

where α_{ij} denote the level of attention from node i to neighbor node j for all $j \in N_i$, where N_i represent the neighborhood of node i.

6.2 Training step

Figure 6.1 shows our proposed attention-based model in the training step. We feed the data collected in a time instance of the set $\{t_1, t_2, ..., t_p\}$ for each learning epoch. $\mathbf{X}$ represents the collection of vectors of each $j \in [2, n]$ (assuming that node 1 was selected for testing). Each X_j vector is composed of meteorological variables, NDVI, NTL, and DEM. Moreover, the structured graph for each one is built based on the selected k. The Union of structured graphs and the $PM_{2.5}$ concentrations of neighbor nodes are weighted by Normalized attention weights for each i computed by Equation 6.5 and based on the attention weight calculated by the attention kernel selected. Then, the weighted linear combinations with their respective features are joined. Further, the activation function (LeakyReLU) is applied to get the sum results are the transformed feature representation for each node, which are entered to FCNN to predict $PM_{2.5}$ concentrations for each one.

6.3 Testing step

In each learning epoch, we evaluate our trained model using the node selected for the testing phase. Similarly to the training step, we obtained the transformed feature representation, which is entered to FCNN to obtain the air pollutant concentration for this node. This process is illustrated in Figure 6.2.

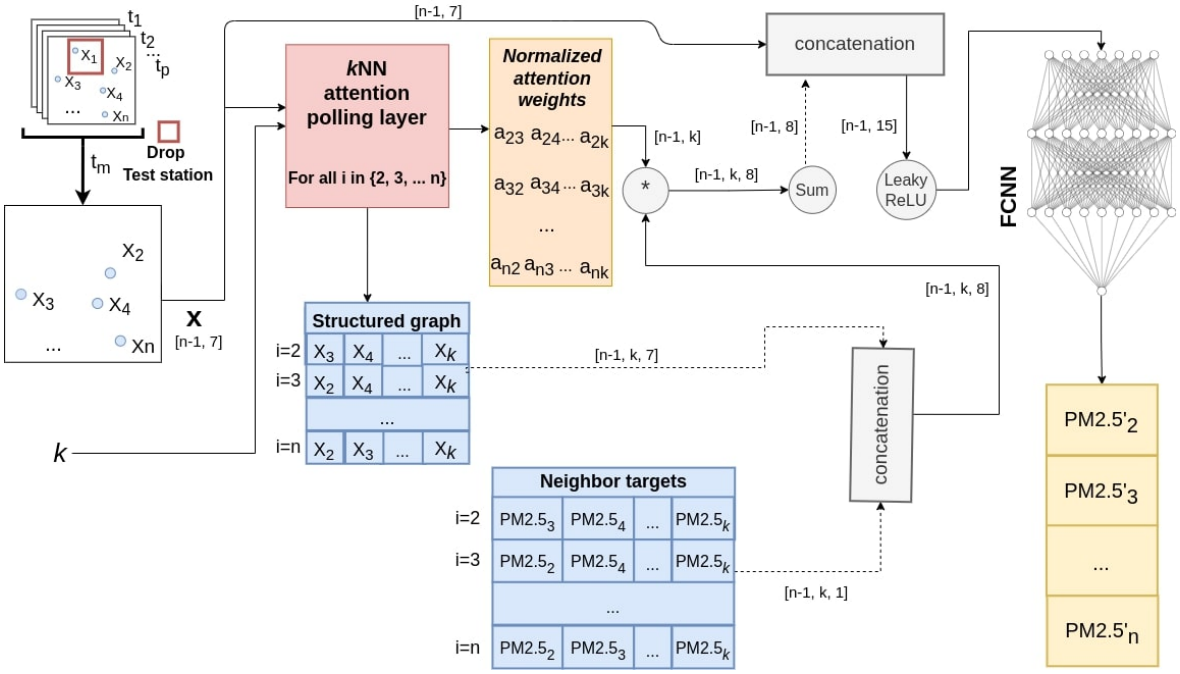


Figure 6.1: kNN attention polling layer and FCNN for spatial prediction of $PM_{2.5}$ concentrations in training step

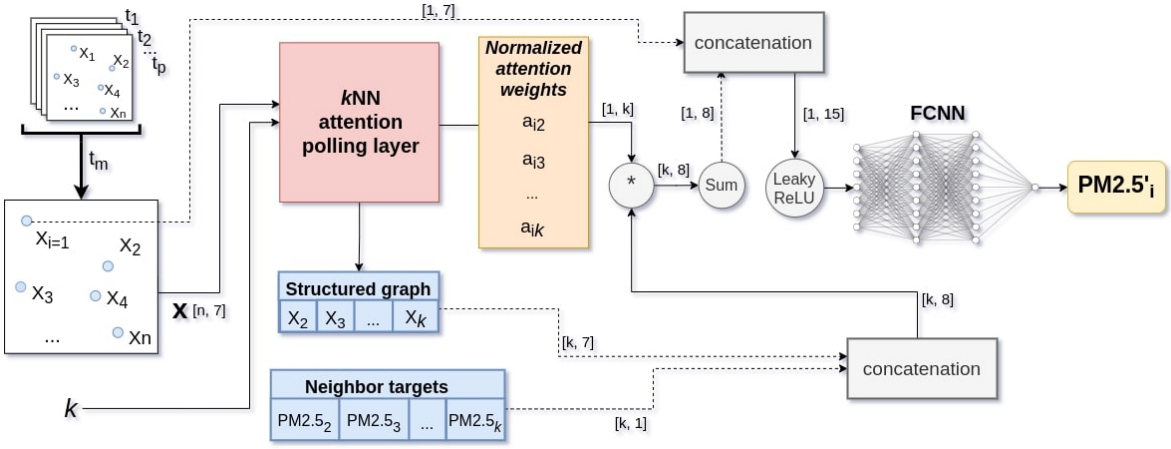


Figure 6.2: kNN attention polling layer and FCNN for spatial prediction of $PM_{2.5}$ concentrations in testing step

Chapter 7

Experimentation I: Data Beijing

For the present experiment, we used data with information on $PM_{2.5}$ concentrations collected by 12 AQM stations distributed on some points of the urban area of Beijing recollected from 01-01-2015 to 31-12-2016. Figure 7.1 shows the pollutant concentrations and statistics of $PM_{2.5}$ observed by the sensors implemented in these stations in the period mentioned.

Figure 7.1: Statistics of Fine Particulate Matter observed by 12 Air Quality Monitoring stations of the Beijing Network in 2015 and 2016. Source: Own

We considered auxiliary meteorological data, such as T, RH, P, WS and WD observed for the same locations of AQM stations. In addition, the satellite information (NDVI, NTL and DEM).

The data was cleaned and prepared for the experimentation phase based on the process explained in Section 4.1. Thus, we got cleaned data for each grid with labeled (points where an AQM station is available) or unlabeled data (points without the observed value of pollution). We split the labeled data (information of 11 AQM stations) for the training step to fit our model, and one AQM station was established as the testing station to evaluate our model. We repeated this process until all AQM stations were testing stations once.

The models implemented and evaluated were the proposed cGAN and an extension called cGANSL (Chapter 5), as well as a Neural Network with an attention-based layer (Chapter 6). Additionally, other models were implemented as IDW and OK as traditional interpolation models. All models were adjusted using labeled data. However, only cGANSL, which is the extension that added a spatial loss calculation to the adversarial learning process of cGAN is capable of handling unlabeled data for fitting. Therefore, this is the only model that uses labeled and unlabeled data.

7.1 Analysis of $PM_{2.5}$ data in AQM station of Beijing

Before the experimentation of the models for spatial prediction of $PM_{2.5}$, we analyze the $PM_{2.5}$ data observed in AQM stations to show the relationship from one to others. Figure 7.2 shows the correlation map based on a scatter plot of the time series observation of $PM_{2.5}$ concentrations. There are high correlations between the data of the AQM stations, which are longer than 0.8.

7.2 Results of traditional interpolation models using Beijing data

7.2.1 Results of Inverse Distance Weighting

We evaluate IDW for the spatial prediction of $PM_{2.5}$ using one AQM station as testing data and one of three different k parameters $\{3, 5, 7\}$ for each experiment. Additionally, we got better results using parameter $p = 1$ established on IDW. Figure 7.3 shows the results of the IDW considering the different configuration and based on RMSE and MAE error metrics.

Table 7.1 shows the average results of IDWusing testing stations and for each k parameter based on the metrics RMSE, MAE and R2. Therefore, we found the best results by inferring the concentration levels of $PM_{2.5}$ by IDW at unobserved points using the information from its three closest AQM stations ($k = 3$).

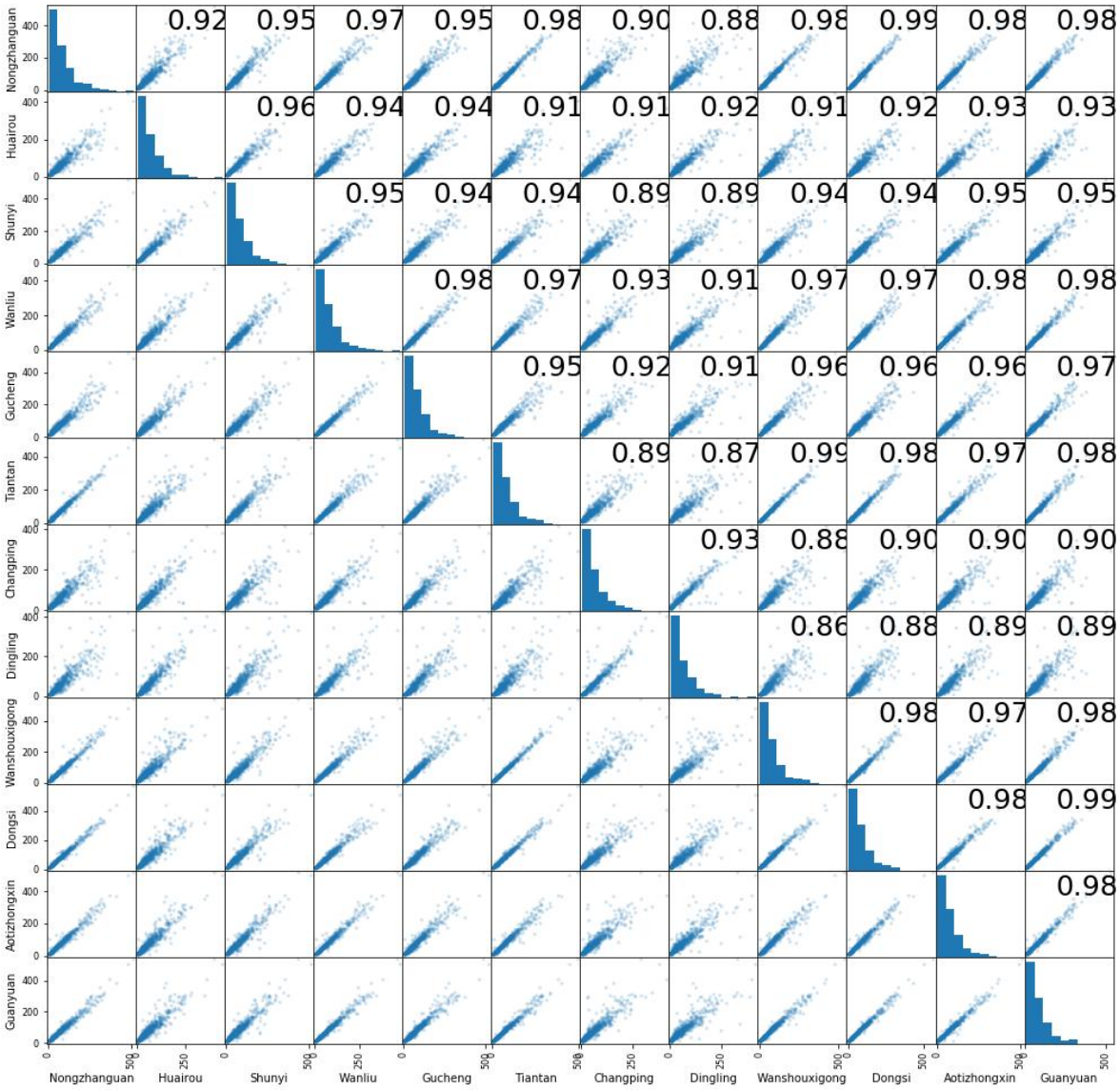


Figure 7.2: Relationship between Air Quality Monitoring stations of the Beijing Network. Source: Own

Metrics	**k=3**	**k=5**	**k=7**
RMSE (μgm^{-3})	**13.0735**	13.3930	14.1413
MAE (μgm^{-3})	**7.5830**	7.8231	8.3189
R2	**0.9549**	0.9511	0.9439

Table 7.1: Average of metrics obtained by IDW using all testing stations, with different values of the kNN parameter ($k = \{3, 5, 7\}$).

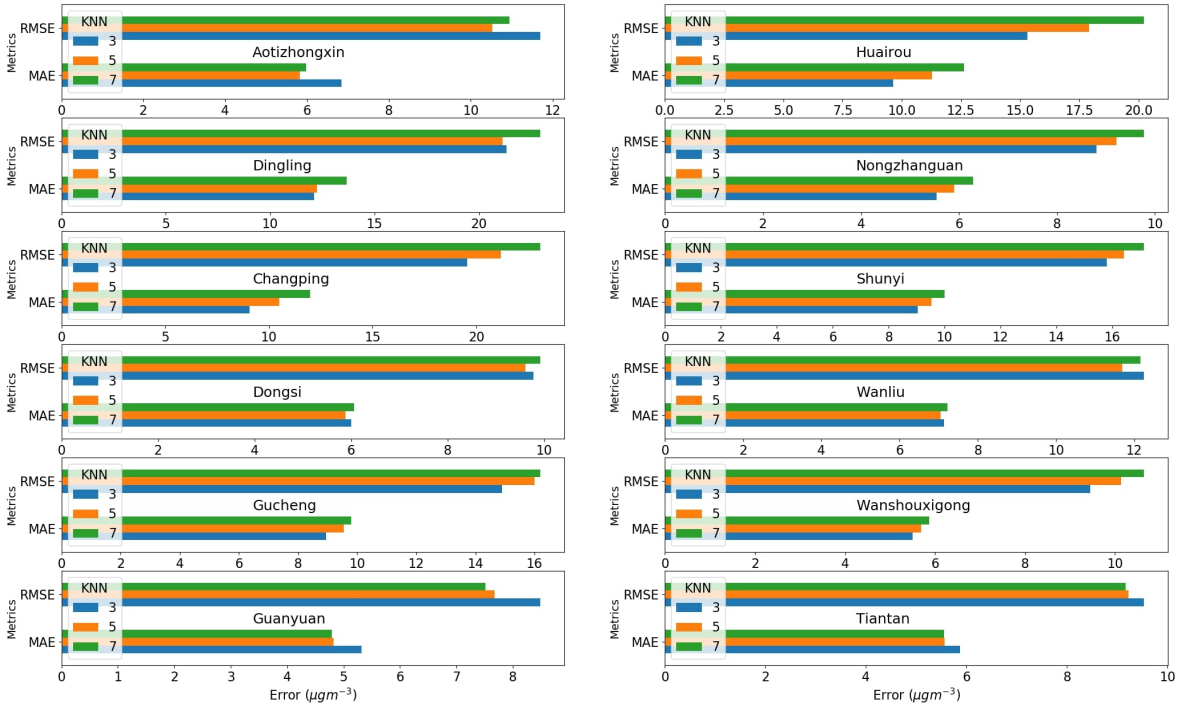


Figure 7.3: Results of Inverse Distance Weighting for each testing station and different settings of the k parameter. Source: Own

7.2.2 Results of Ordinary Kriging

We evaluated OK for the spatial prediction of $PM_{2.5}$ using testing data as an AQM station and a Variogram established in OK for each experiment. The Variograms considered for the experiment are linear, power, gaussian, spherical, exponential and hole-effect. Figure 7.4 shows the results of the OK based on the error metrics RMSE and MAE.

Table 7.2 shows the average of results of the experimentation with all testing stations for each Variogram, based on three metrics RMSE, MAE and R2. We found the best results by setting Variogram to Linear.

Metrics/Variog.	**Linear**	**Power**	**Spherical**	**Gaussian**	**Exponen.**	**Hole-effect**
RMSE (μgm^{-3})	**12.8640**	13.0313	12.8921	22.8024	12.9912	17.6750
MAE (μgm^{-3})	**7.4216**	7.5330	7.5010	12.3945	7.6280	10.4412
R2	**0.9555**	0.9549	0.9544	0.8373	0.9538	0.9151

Table 7.2: Average of metrics obtained by OK using all testing stations for each variogram setup (Linear, Power, Spherical, Gaussian, Exponential and Hole-Effect).

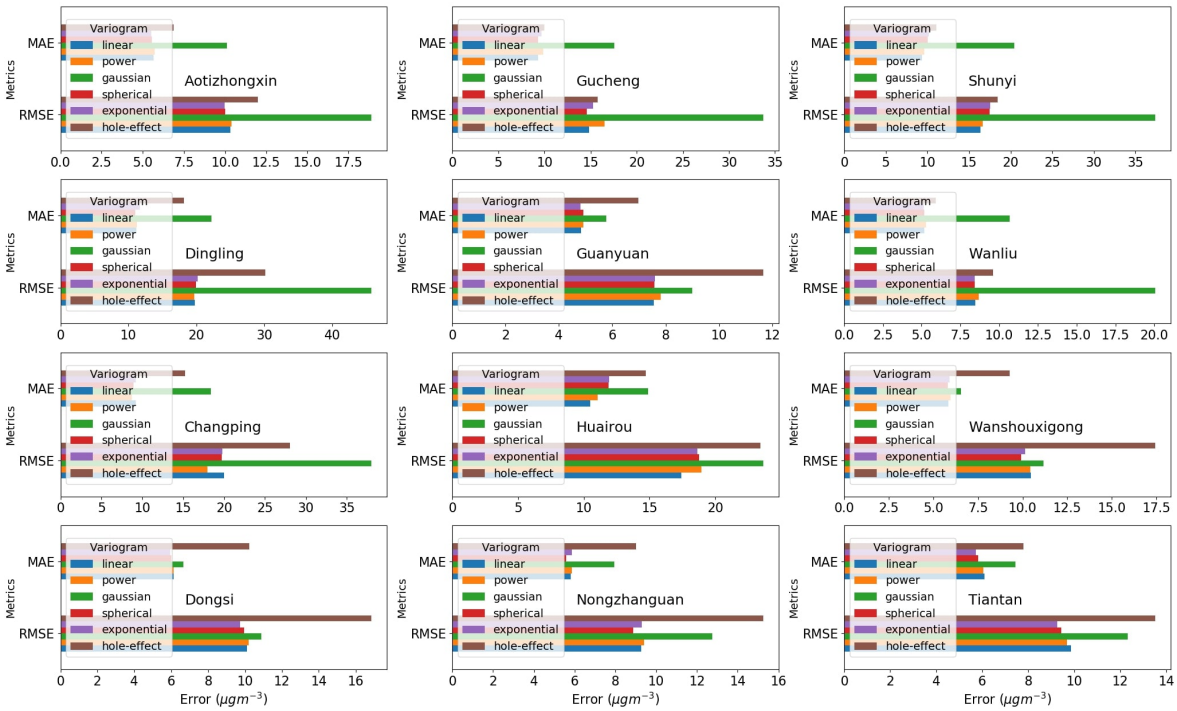


Figure 7.4: Results of Ordinary Kriging for each testing station and different settings of Variogram. Source: Own

7.3 Results of the method I: cGAN and cGANSL using Beijing data

cGAN is trained by adversarial learning and using labeled data. Figure 7.5 shows the confrontational-training process between G and D. At the first learning iterations, the error of G is smaller than the error of D because D accepts fake samples generated by G, due to this is not yet fitted and does not discriminate well.

For each training epoch, we evaluate cGAN using testing data (data from the AQM station selected as the testing station) and according to different kNN parameters (k), based on the three metrics RMSE, MAE and R2. Figure 7.6 shows the evolution of RMSE in training process for each k value and each testing station.

When cGAN managed to stabilize, we obtained the best results for each k configuration based on the average of the metrics reached by the stations, which is shown in Table 7.3. The configuration $k = 3$ was the best choice for this model.

Furthermore, we proposed and evaluated cGANSL, which was explained in Section 5.2. To evaluate the cGANSL, we established the parameters θ and λ, which represent the parameter of spatial learning and adversarial learning. The spatial parameter offsets the training learning based on adversarial learning. We experimented with different combination of values for α and θ, set the kNN parameter to 3 ($k = 3$).

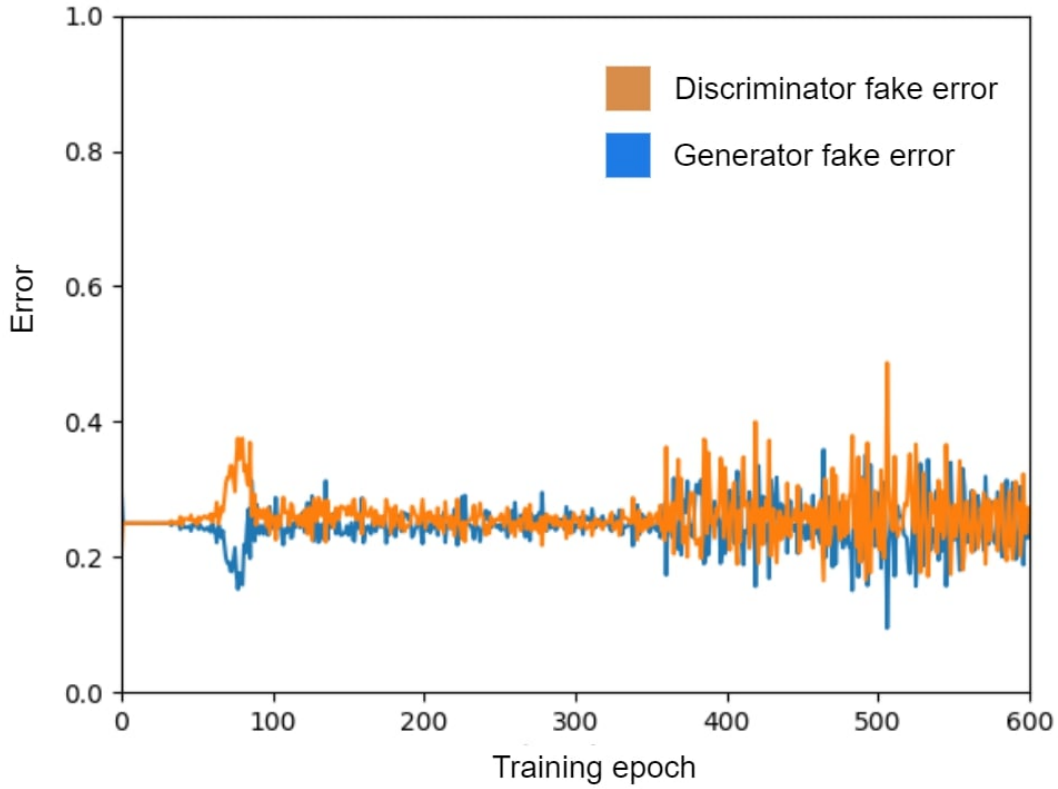


Figure 7.5: Fake error of Discriminator vs fake error of Generator. Source: Own

Figure 7.7 shows the RMSE of the training process of cGANSL using each combination of values and for each testing station.

Metrics	**k=3**	**k=5**	**k=7**
RMSE (μgm^{-3})	**14.2050**	18.5328	15.9306
MAE (μgm^{-3})	**8.8692**	12.3294	9.8941
R2	**0.9488**	0.8683	0.9374

Table 7.3: Average of metrics obtained by cGAN using all test stations, with different values of the kNN parameter ($k = \{3, 5, 7\}$).

Table 7.4 shows the average of the metrics obtained for each value combinations results of cGANSL. The best combination of parameters for cGANSL was ($\lambda = 1, \theta = 2$) according to the metrics RMSE and R2. However, on the metric MAE the best combination of parameters was ($\lambda = 0, \theta = 1$). Moreover, similar to IDW, the best setting for the parameter k was 3 (k=3) for cGAN and cGANSL.

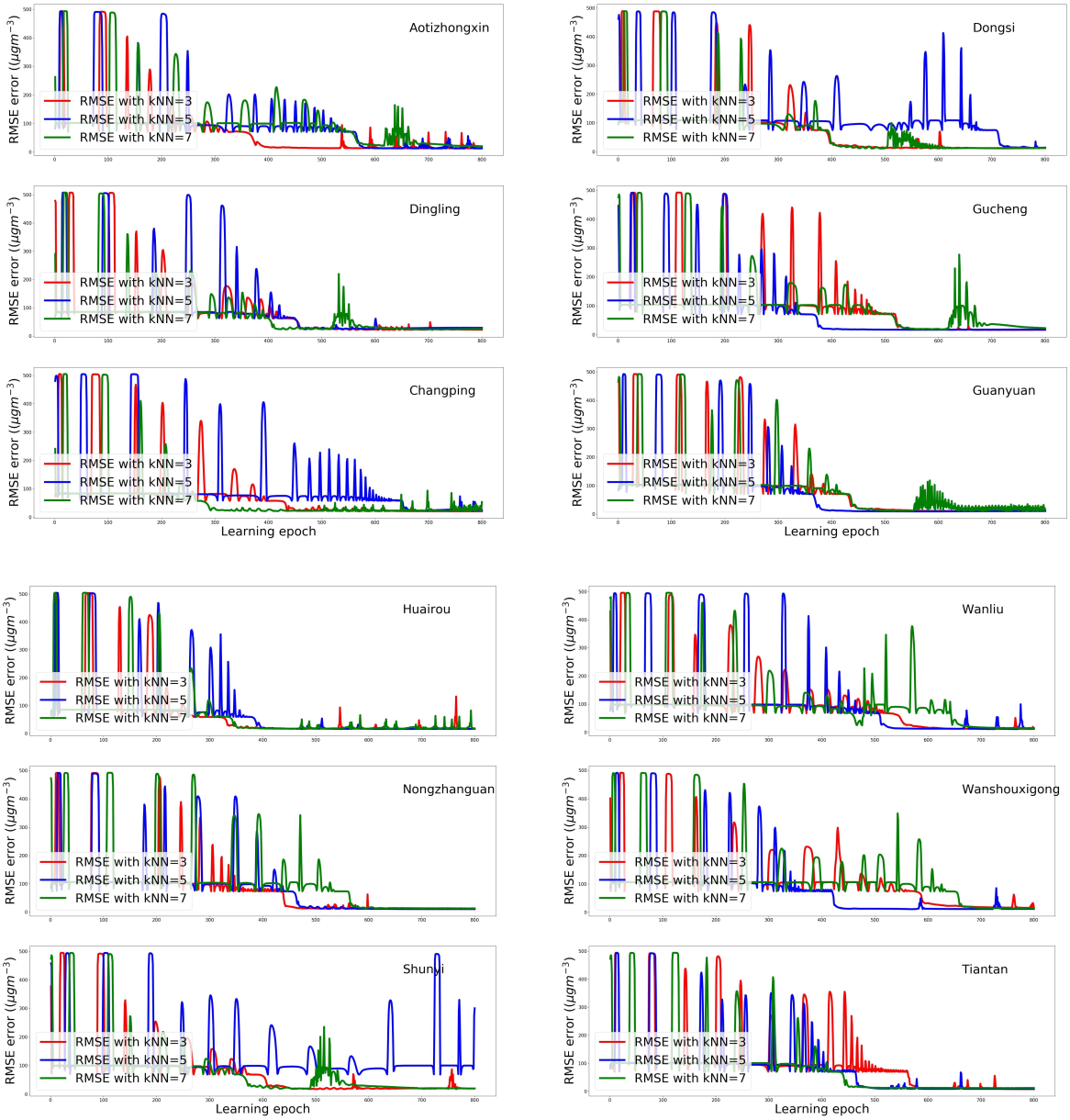


Figure 7.6: RMSE of cGAN in its learning process for each testing station and different setting of k parameter. Source: Own.

Metrics / Comb.(λ, θ)	**(0,1)**	**(0.1,0.9)**	**(0.2,0.8)**	**(0.4,0.6)**	**(1,2)**	**(2,1)**
RMSE (μgm^{-3})	12.9104	12.7609	12.8744	12.8807	**12.6862**	13.1295
MAE (μgm^{-3})	**7.4931**	7.6109	7.7048	7.7189	7.7254	8.1085
R2	0.9552	0.9575	0.9565	0.9559	**0.9577**	0.9547

Table 7.4: Average of metrics obtained by cGANSL using all testing stations, with different values of Adversarial learning (λ) and Spatial learning (θ) parameters, as well as kNN parameter set to 3 ($k = 3$)

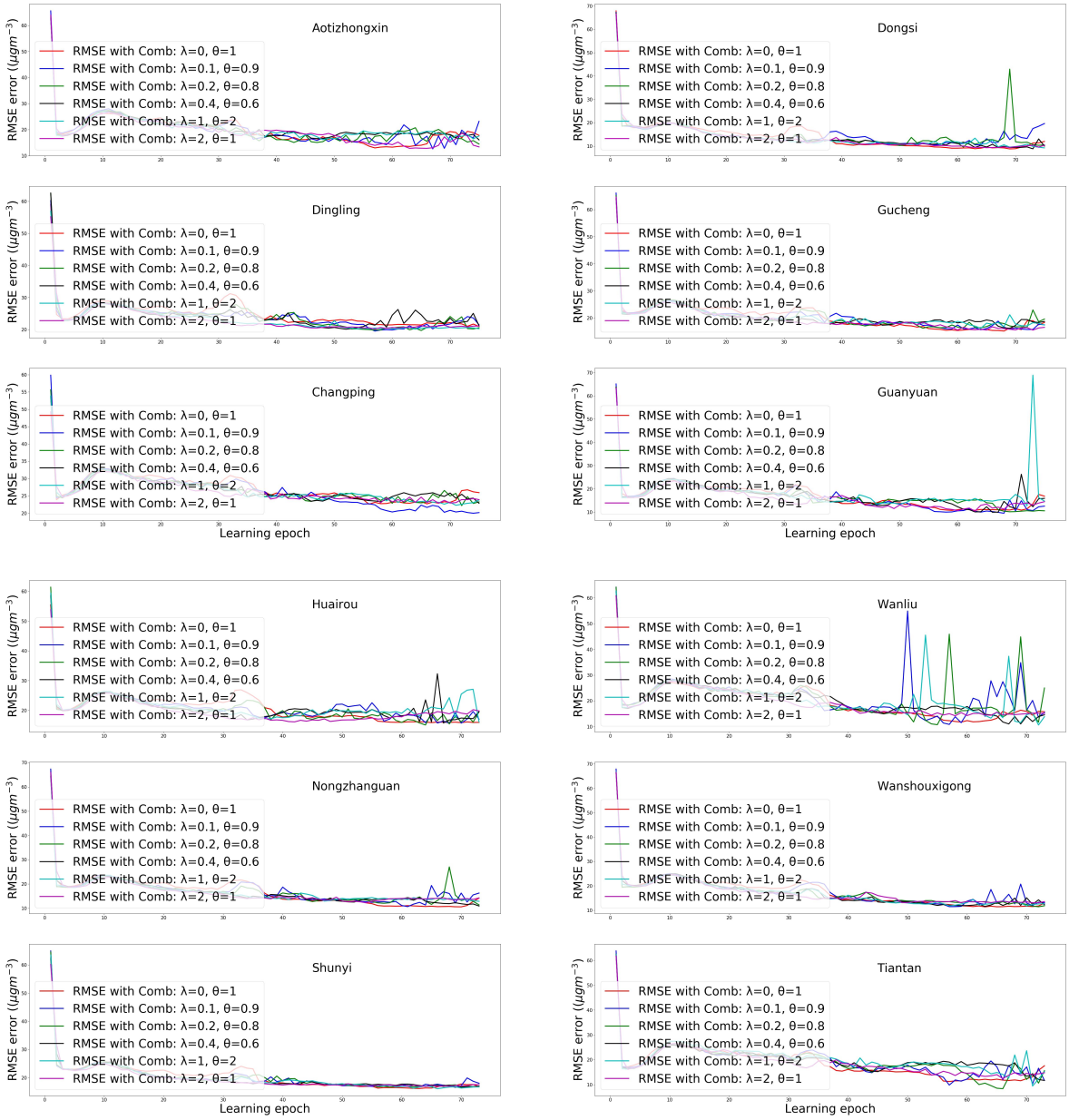


Figure 7.7: RMSE of cGANSL in its learning process using different parameter combinations (α,θ) and for each Air Quality monitoring station as testing data. Source: Own.

7.4 Results of method II: Neural Network with an attention-based layer using Saõ Paulo Data

In this section, we present the experimental results of the spatial prediction of $PM_{2.5}$ concentrations by a kNN attention polling layer stacked with FCL2. This model is explained in detail in Chapter 6.

The model is trained and tested using labeled data. We choose for the training step the data obtained from training stations. Then we applied the model shown in Figure 6.1. After we applied the model shown in Figure 6.2 using data obtained from one station selected for the testing step (which was not included in the training step).

For each testing station and many learning epochs, we trained Neural Network with an attention-based layer. We adjusted the model with training data to evaluate the performance with testing data in each learning epoch. Figure 7.8 shows the performance of our model for each Attention Kernel.

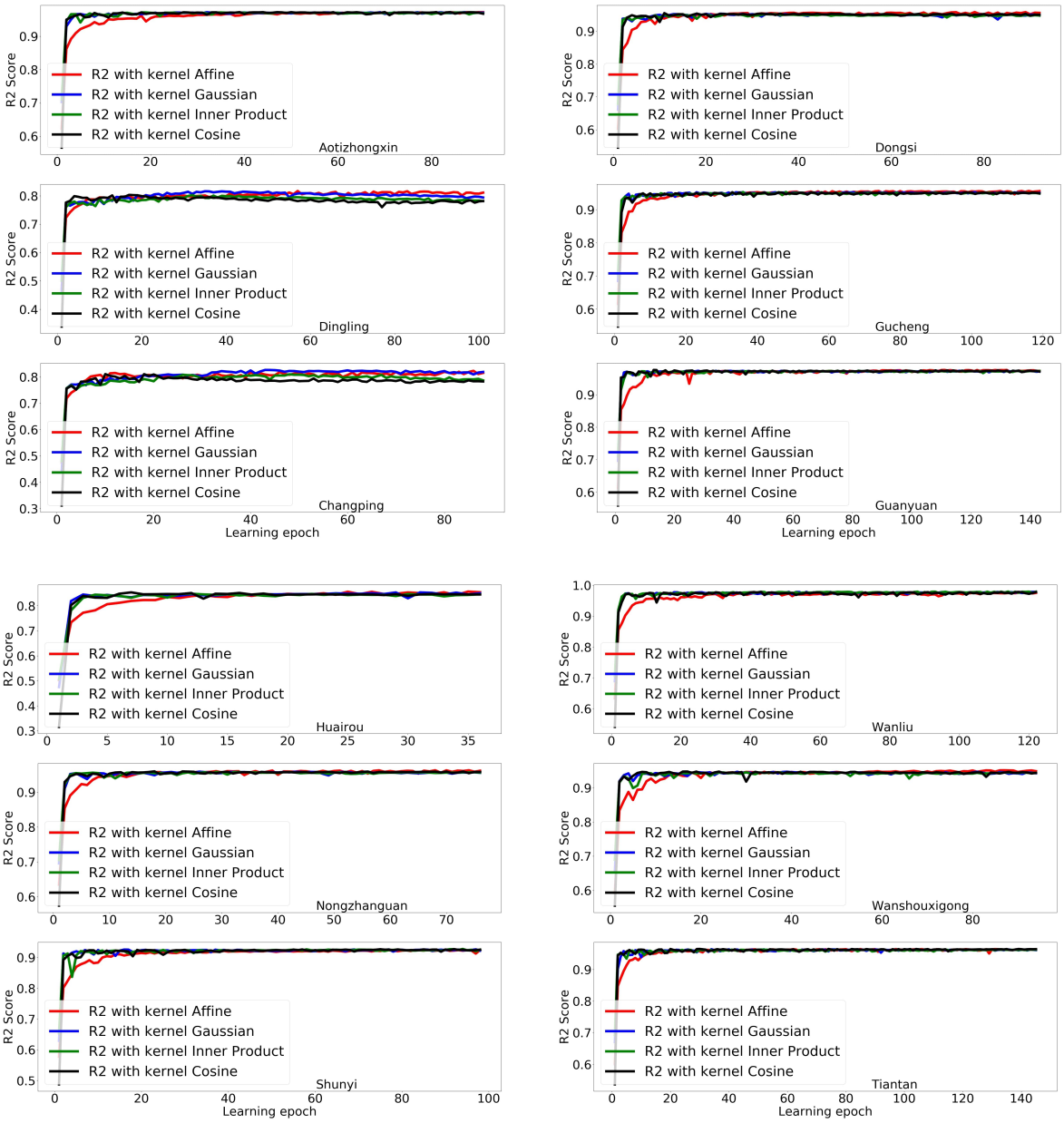


Figure 7.8: R2 Score of the Neural Network with an attention-based layer using testing data in its learning process. Source: Own.

Table 7.5 shows the average of the results of the testing stations for each Attention Kernel.

Metrics	Inverse distance	Inner-product	Perceptron affine	Cosine
RMSE (μgm^{-3})	16.4326	16.8254	**16.1124**	16.7995
MAE (μgm^{-3}	9.7561	9.9888	**9.5107**	10.1792
R2	0.9291	0.9244	**0.9308**	0.9238

Table 7.5: Results of the attention-based layer with each kernel attention using testing data.

We got better results by setting the Attention Kernel as 'Perceptron Affine' in our model, based on the metrics RMSE, MAE and R2.

7.5 Comparison of all models using data of Beijing Network

7.5.1 Performance of Spatial-Prediction Models

In this section, we compare all tested models with their best-found fits using data from AQM stations of the Beijing Network. Table 7.6 shows the average results of prediction models.

Metrics	**IDW**	**OK**	**cGAN**	**cGANSL**	**NN&Atten. Layer(Affine)**
RMSE (μgm^{-3})	13.0735	12.8640	14.2050	**12.6862**	16.1124
MAE (μgm^{-3})	7.5830	7.4216	8.8692	**7.7254**	9.5107
R2	0.9549	0.9555	0.9488	**0.9577**	0.9308

Table 7.6: Result averages of results of the testing stations using spatial prediction models.

Figure 7.9 shows the $PM_{2.5}$ concentrations observed by AQM stations of Beijing from 2015-05-01 to 2015-05-30 and prediction outputs of the fitted models.

7.5.2 Region Beijing Map of estimated Fine Particulate Matter by Spatial-Prediction Models

Figure 8.14 shows the heat maps of observation of $PM_{2.5}$ by AQM stations in the Beijing network, predicted $PM_{2.5}$ concentrations on locations of Beijing by spatial

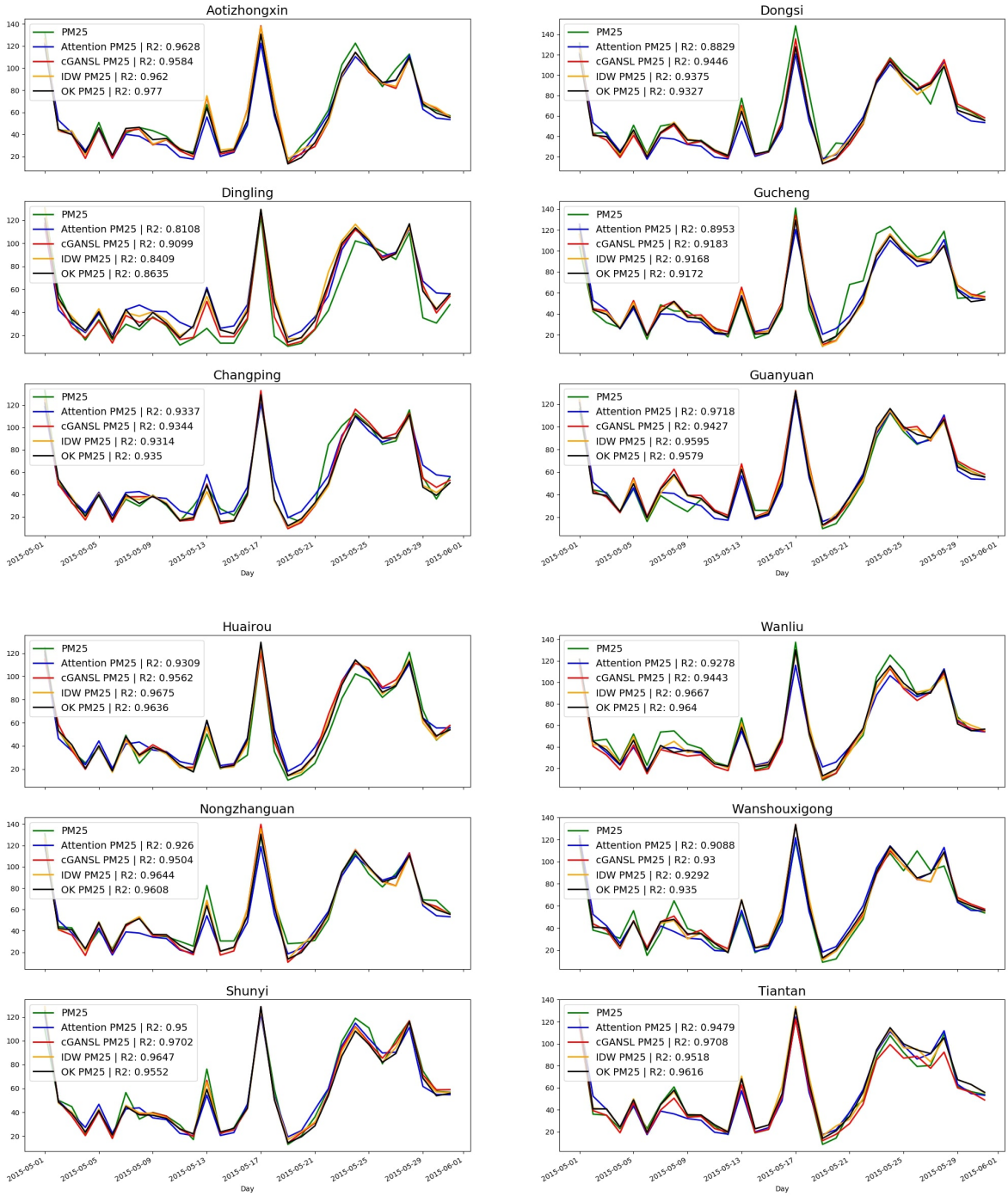


Figure 7.9: Outputs of spatial prediction models and observed values at air quality monitoring stations of Beijing Network from 2015-05-01 to 2015-05-30. Source: Own.

prediction models, as well as vegetation and population levels, infer from NDVI and NTL, respectively.

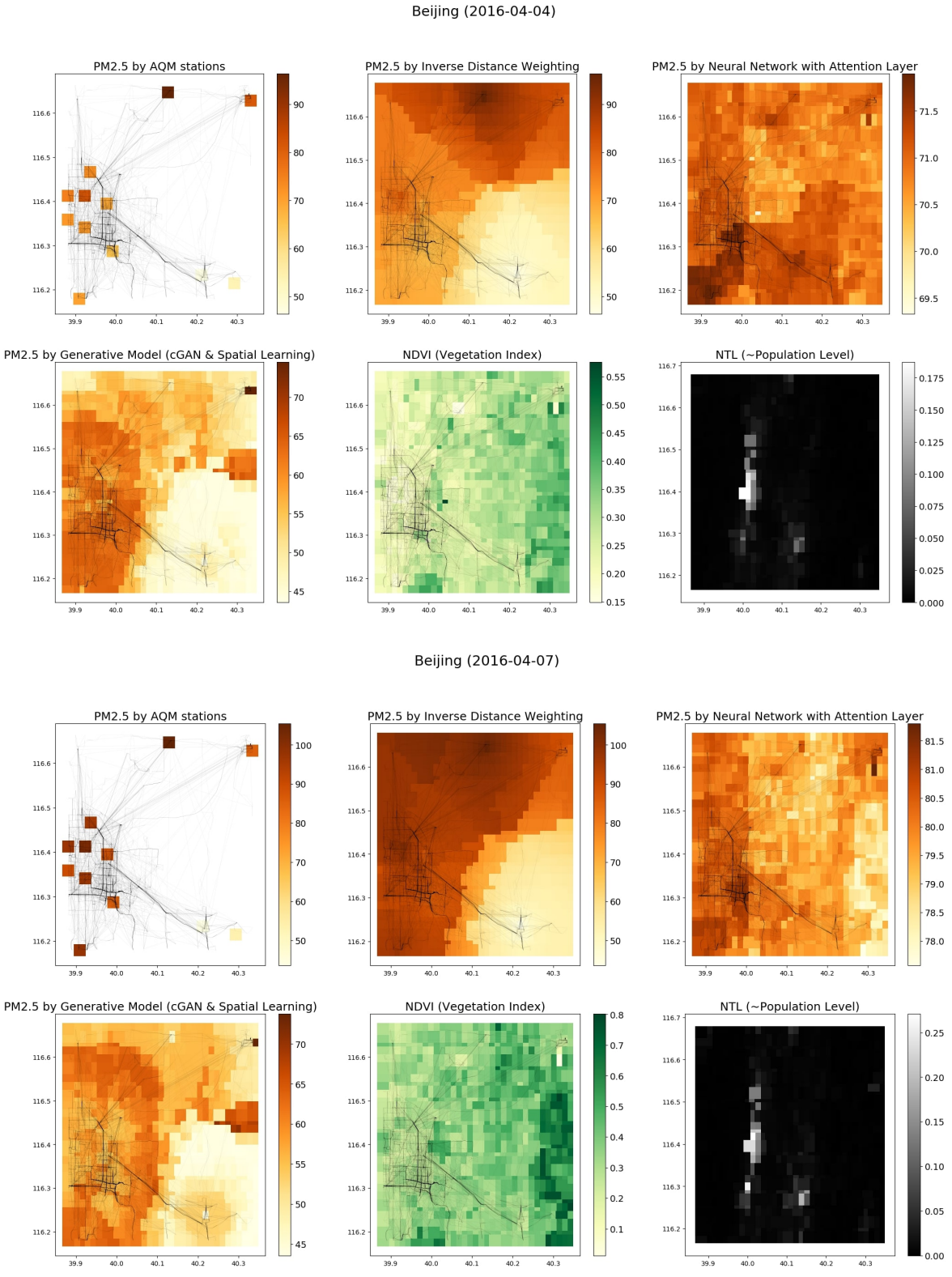


Figure 7.10: Heat maps of outputs of prediction models, observed concentrations of fine particulate, Normalized Difference Vegetation Index, and Nightime index, which infer the population level in some locations of Beijing. Source: Own.

Chapter 8

Experimentation II: São Paulo Data

In this chapter, we present the experimentation using data with information on $PM_{2.5}$ concentrations monitored by 11 AQM stations distributed on some points of the urban area of São Paulo collected from 2017-01-01 to 2019-12-31. Figure 8.1 shows the time series of air pollutant concentrations by each AQM station and the correlation between them in this region.

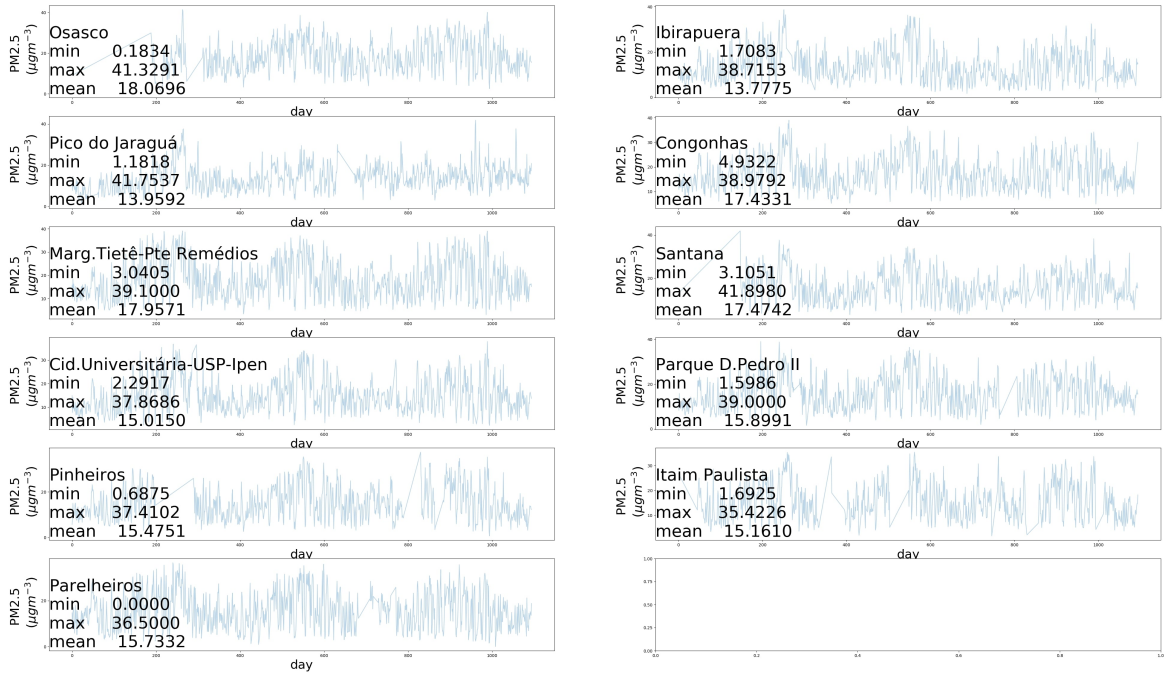


Figure 8.1: Statistics of $PM_{2.5}$ observed in AQM stations of São Paulo. Source: Own

We used data obtained from AQM stations and the nearest meteorological monitoring station, sparsely distributed in a region of São Paulo city. The data are $PM_{2.5}$ concentrations and meteorological conditions (T, P, RH, WD). Moreover, land-related variables are recollected from satellite products (explained in Section 2.4). Then, all

obtained data was preprocessed and matched based on the methods detailed in Section 4.1.

After the preprocessing and matching step, the result data is the information from each grid of the divided region of São Paulo. For each grid, we have meteorological conditions, land-related variables, population-related variables and the $PM_{2.5}$ observed if a monitoring station of air quality is available, which are labeled data and otherwise, unlabeled data (Section 4.1). Also, for each grid, we have the result data for each day from 01/01/2017 to 12/31/2019.

In a similar way that the experimentation with Beijing data in Chapter 7, we implemented IDW, OK and cGAN and our proposed models cGANSL and Neural Network with Attention-based layer. The models except cGANSL considered only labeled data for its training and testing step. cGANSL is capable of handling unlabeled data.

8.1 Analysis of $PM_{2.5}$ data in AQM station of São Paulo

Figure 8.2 shows the correlation between the AQM stations of São Paulo. The correlation of these stations is very low compared with the correlation of the AQM stations of Beijing data showed in Figure 7.2.

8.2 Results of traditional interpolation models using São Paulo data

8.2.1 Results of Inverse Distance Weighting

We show the results of IDW for the spatial prediction of $PM_{2.5}$ in São Paulo. For each experimentation, we use a AQM station selected as the testing station and one of four different k parameters 3, 5, 7, 10. Moreover, we established $p = 1$ on this interpolation model. Figure 8.3 shows the results of the IDW experimentation, based on two error metrics: RMSE and MAE.

Table 8.1 shows the average of the IDW results using testing stations and for each k parameter based on the metrics RMSE, MAE and R2. The best results by inferring the concentration levels of $PM_{2.5}$ by IDW at unobserved points using the information from its 10 closest AQM stations ($k = 10$).

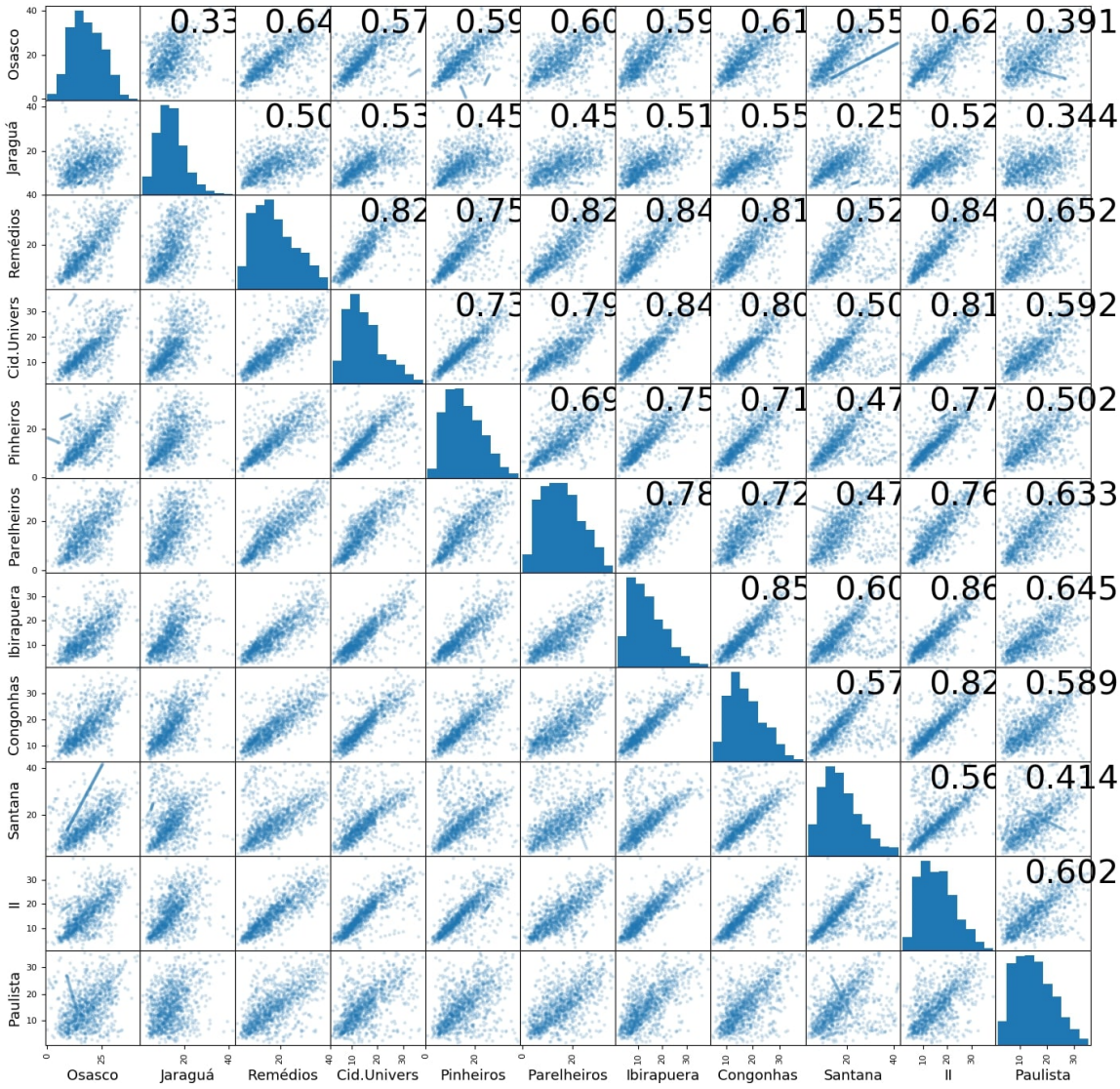


Figure 8.2: Relationship between Air Quality Monitoring stations of the São Paulo Network. Source: Own

Metrics	**k=3**	**k=5**	**k=7**	**k=10**
RMSE (μgm^{-3})	5.4591	5.1613	5.0289	**4.9339**
MAE (μgm^{-3})	3.8630	3.6497	3.5652	**3.4905**
R2	0.3949	0.4540	0.4824	**0.5008**

Table 8.1: Average of metrics obtained by IDW using all testing stations, with different values of the kNN parameter ($k = \{3, 5, 7\}$).

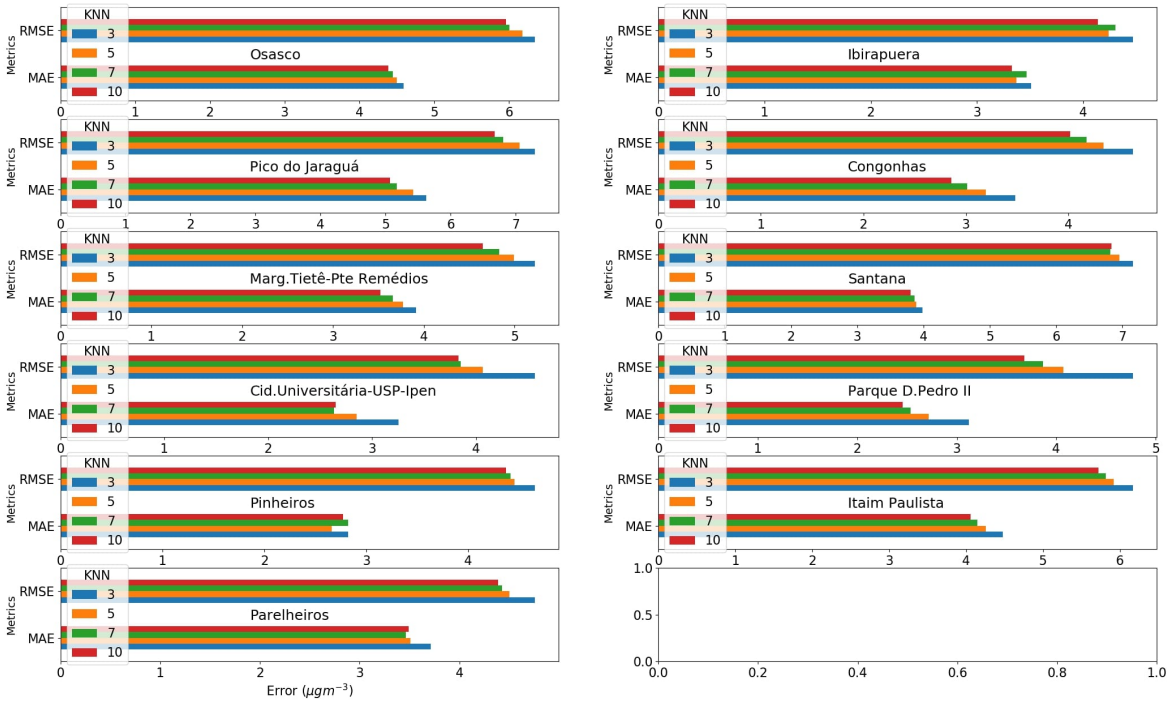


Figure 8.3: Results of Inverse Distance Weighting for each testing station and different settings of the k parameter using São Paulo data. Source: Own

8.2.2 Results of Ordinary Kriging

Using OK for the spatial prediction of $PM_{2.5}$, testing data as an AQM station and a Variogram (linear, power, gaussian, spherical, exponential and hole-effect) established on OK for each experiment we found the results based on the error metrics RMSE and MAE shown in Figure 8.4.

Table 8.2 shows the average of results of the experimentation with all testing stations for each Variogram, based on three metrics RMSE, MAE and R2. We found the best results by setting Variogram to Linear.

Metrics/Variog.	**Linear**	**Power**	**Spherical**	**Gaussian**	**Exponen.**	**Hole-effect**
RMSE (μgm^{-3})	4.9280	4.9956	5.0791	13.4908	5.0420	**4.8891**
MAE (μgm^{-3})	3.5196	3.5702	3.6054	5.9819	3.5764	**3.4772**
R2	0.4964	0.4833	0.4733	-8.3939	0.4811	**0.5124**

Table 8.2: Average of metrics obtained by OK using all testing stations for each variogram setup (Linear, Power, Spherical, Gaussian, Exponential and Hole-Effect) using São Paulo data.

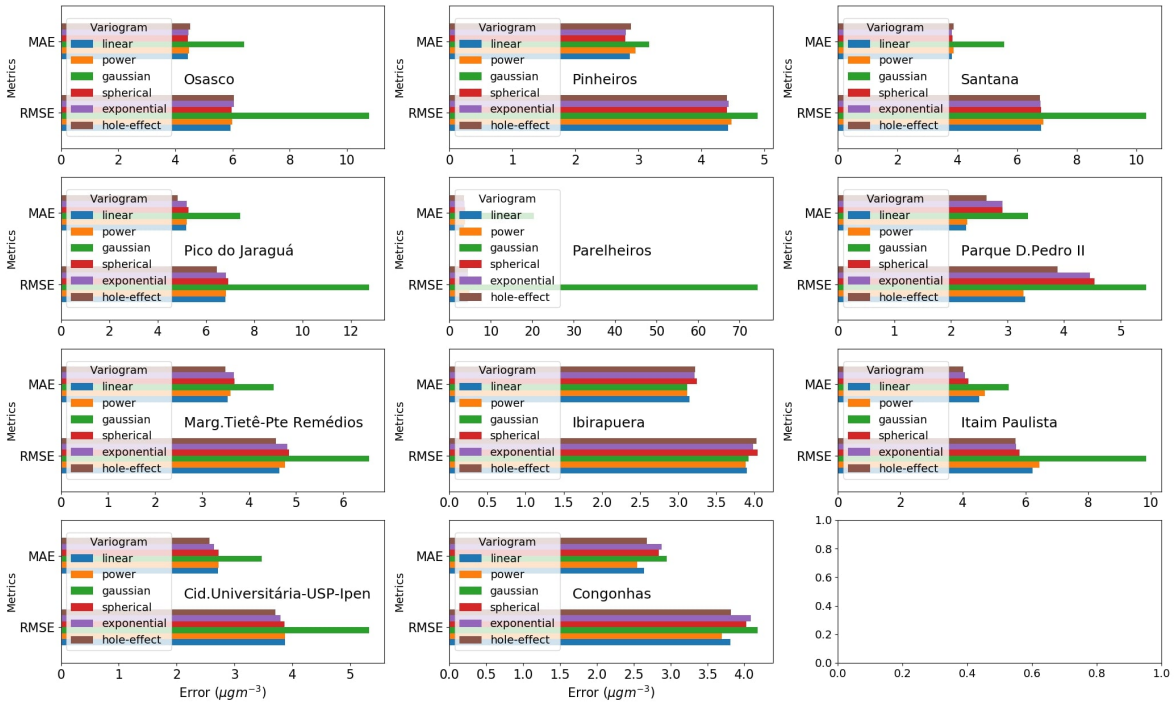


Figure 8.4: Results of Ordinary Kriging for each testing station and different settings of Variogram using São Paulo data. Source: Own

8.3 Results of the method I: cGAN and cGANSL using São Paulo data

cGAN is trained by adversarial learning and using labeled data. In each training epoch, we evaluate cGAN using different parameters of nearest neighbor station selection $\{3, 5, 7, 10\}$ and testing data (data from the AQM station selected as a testing station), based on the three metrics RMSE, MAE and R2. Figure 8.5 shows the evolution of RMSE in training process for each k value and each testing station. The training process of cGAN using São Paulo data is more unstable than the performance of cGAN on Beijing data (Figure 7.6). However, setting the parameter of nearest neighbor stations as 7 and 10 ($k = \{7, 10\}$), the model stabilized for most testing stations.

Table 8.3 shows the best results for each k configuration reached by the stations, based on the average of the metrics: RMSE, MAE and R2.

Metrics	**k=3**	**k=5**	**k=7**	**k=10**
RMSE (μgm^{-3})	7.1006	6.3197	**5.5564**	6.0171
MAE (μgm^{-3})	5.7366	4.9978	**4.0766**	4.5982
R2	0.0333	0.2040	**0.3798**	0.2658

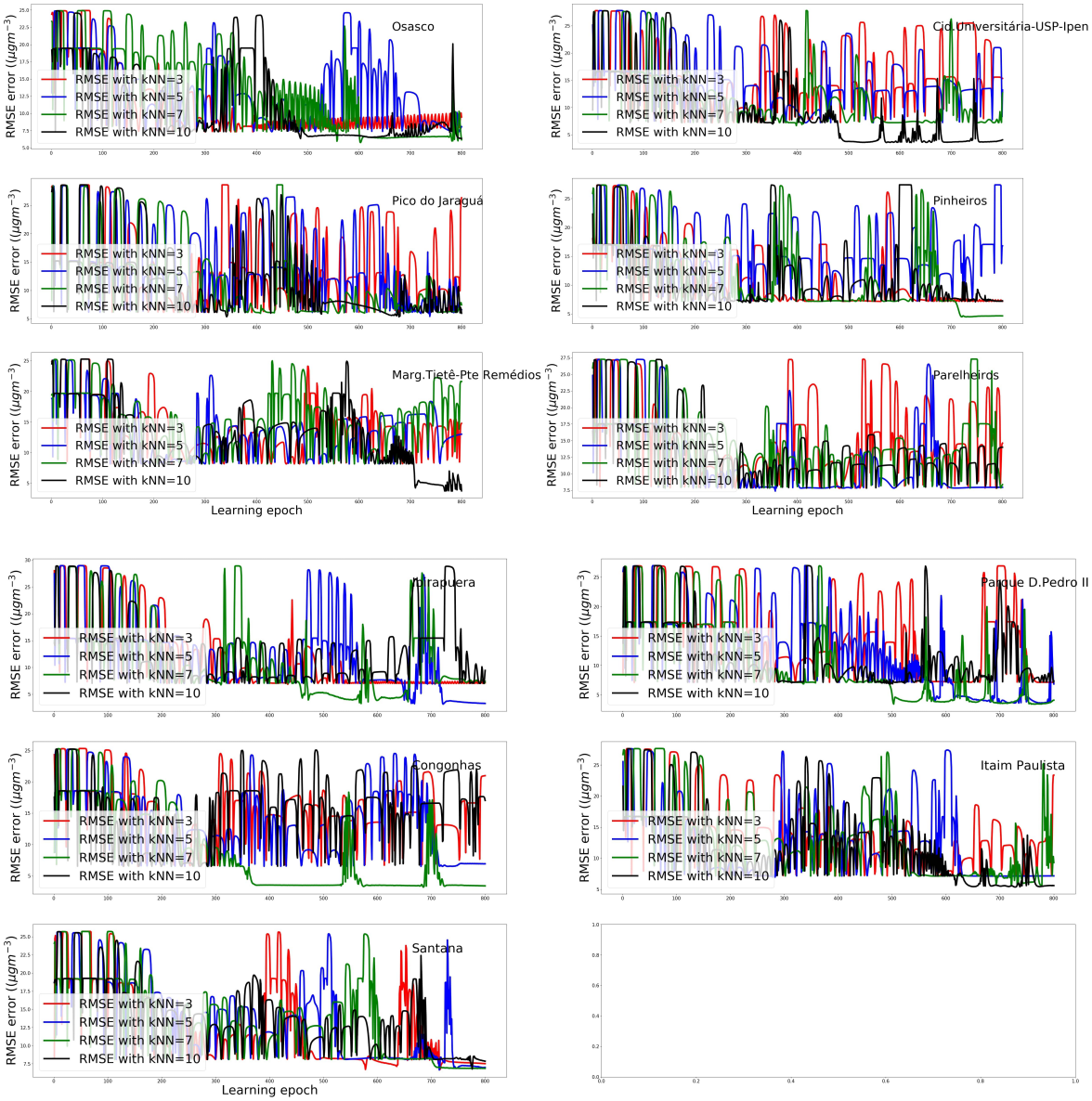


Figure 8.5: RMSE of cGAN in its learning process for each testing station and different setting of k parameter using São Paulo data. Source: Own.

Table 8.3: Average of metrics obtained by cGAN using all test stations, with different values of the kNN parameter ($k = \{3, 5, 7, 10\}$) using São Paulo data.

Considering $k = 10$ as the best nearest neighbor station selection parameter, Figure 8.6 shows the results of cGANSL experimentation using São Paulo data for each AQM station, using different configurations of parameters of adversarial learning (λ) and spatial learning (θ) explained in Section 5.2.

The best combinations based on the result of experimentation was ($\lambda = 0.2, \theta = 0.8$) the average of the metrics obtained for each value combinations results of cGANSL, it is shown in 8.4.

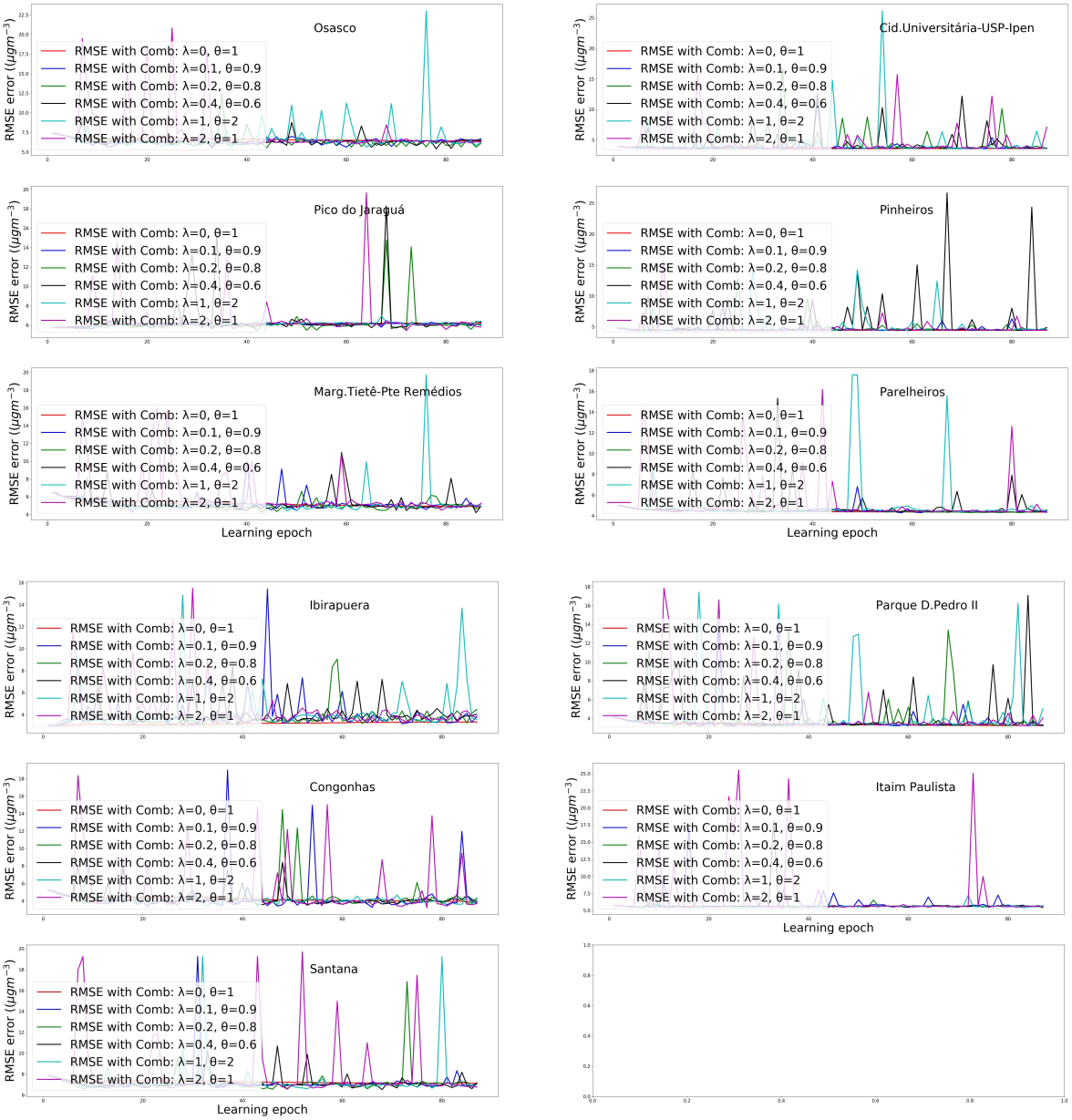


Figure 8.6: RMSE of cGAN in its learning process for each testing station and different setting of k parameter. Source: Own.

Metrics / Comb.(λ, θ)	**(0,1)**	**(0.1,0.9)**	**(0.2,0.8)**	**(0.4,0.6)**	**(1,2)**	**(2,1)**
RMSE (μgm^{-3})	4.7357	4.4759	**4.4478**	4.4791	4.4840	4.5006
MAE (μgm^{-3})	3.2969	3.2191	**3.2185**	3.2188	3.2318	3.2111
R2	0.5407	0.5903	**0.5980**	0.5903	0.5905	0.5875

Table 8.4: Average of metrics obtained by cGANSL using all testing stations, with different values of Adversarial learning (λ) and Spatial learning (θ) parameters, as well as kNN parameter set to 10 ($k = 10$)

8.4 Results of Method II: Neural Network with an attention-based layer using São Paulo data

This section shows the experimentation result of kNN attention polling layer stacked with FCL2 (see Chapter 6) for spatial prediction of $PM_{2.5}$ concentrations using information from AQM stations of São Paulo Network, meteorological variables and land-related data of São Paulo.

The labeled data was used for training and testing the Neural Network with an attention-based layer. In the training step, we applied the model explained in Section 6.2. To test utilized the model detailed in Section 6.3 using data from an AQM station selected for testing. We fit the model based on the number of epochs, and in each learning epoch, we evaluate the performance of this model using data from the testing station.

Figure 8.7 shows the performance of our model for each Attention Kernel.

The best attention kernel for the layer of the model is 'Perceptron Affine' based on the average results of kernel attention on testing stations shown in Table 8.5.

Metrics	Inverse distance	Inner-product	Perceptron affine	Cosine
RMSE (μgm^{-3})	4.5715	4.8106	**4.3317**	4.5474
MAE (μgm^{-3})	3.3137	3.5573	**3.1648**	3.3180
R2	0.5739	0.5379	**0.6295**	0.5785

Table 8.5: Average results of attention kernels on testing stations.

8.5 Comparison of spatial prediction models using data of São Paulo Network

8.5.1 Performance of Spatial-Prediction Models

Table 8.6 shows the average results of tested models. The models are configured with their best-found fits.

Metrics	IDW	OK	cGAN	cGANSL	NN&Atten. Layer(Affine)
RMSE (μgm^{-3})	4.9339	4.8891	5.5564	4.4478	4.3317
MAE (μgm^{-3})	3.4905	3.4772	4.0766	3.2185	3.1648
R2	0.5008	0.5124	0.3798	0.5980	0.6295

Table 8.6: Average results of the spatial prediction models using São Paulo data.

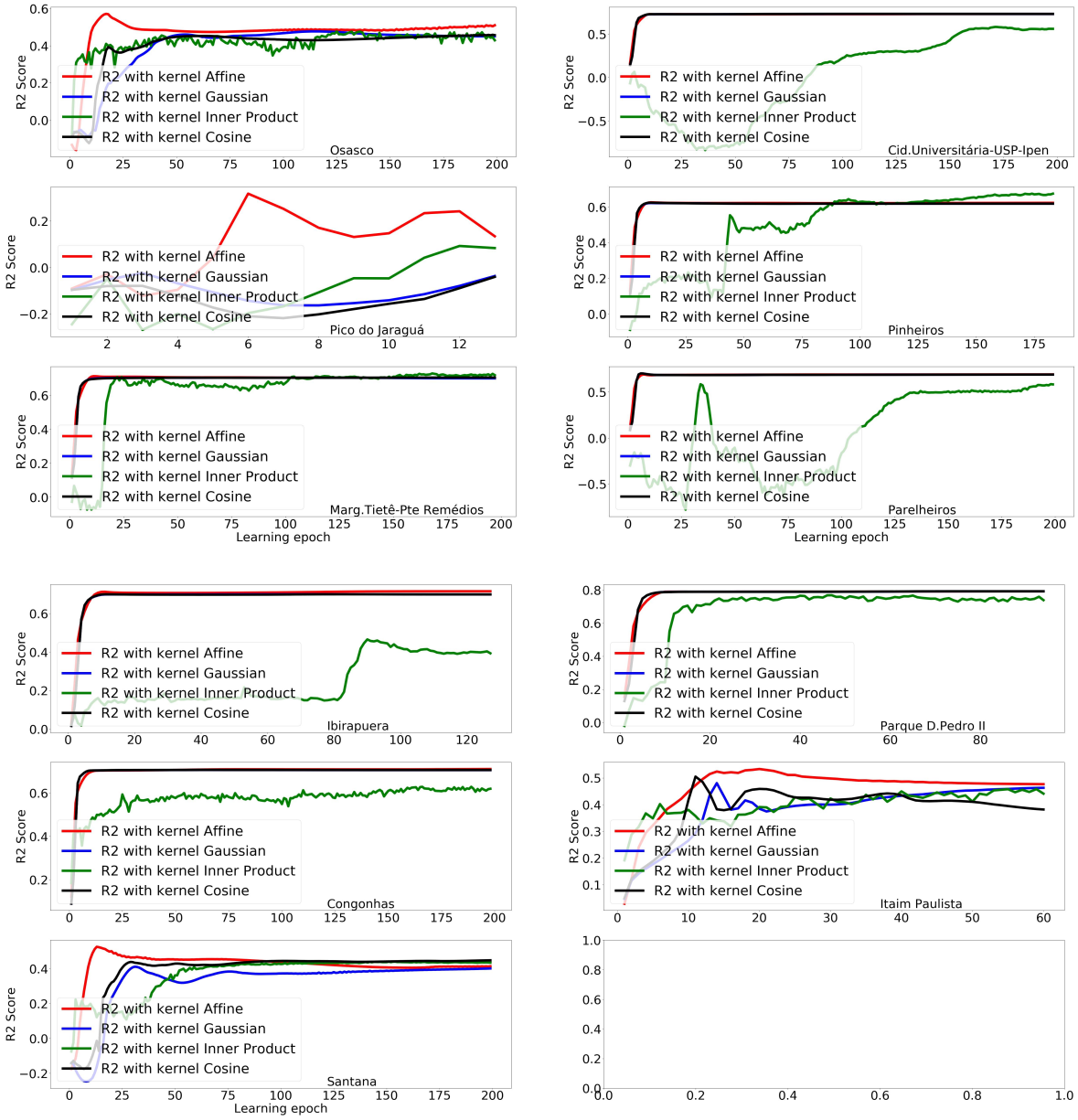


Figure 8.7: R2 Score of the Neural Network with an attention-based layer in its learning process using São Paulo data. Source: Own.

Figure 8.8 shows the $PM_{2.5}$ concentrations observed by AQM stations of São Paulo from 2018-07-01 to 2018-07-31 and prediction outputs of the fitted models.

8.5.2 Attention to the kNN stations from testing station

According to explained in Chapter 6, the kernels of the attention-based layer calculate the normalized attention weights, based on Equation 6.5 for each station in the graph to its nearest neighbor nodes.

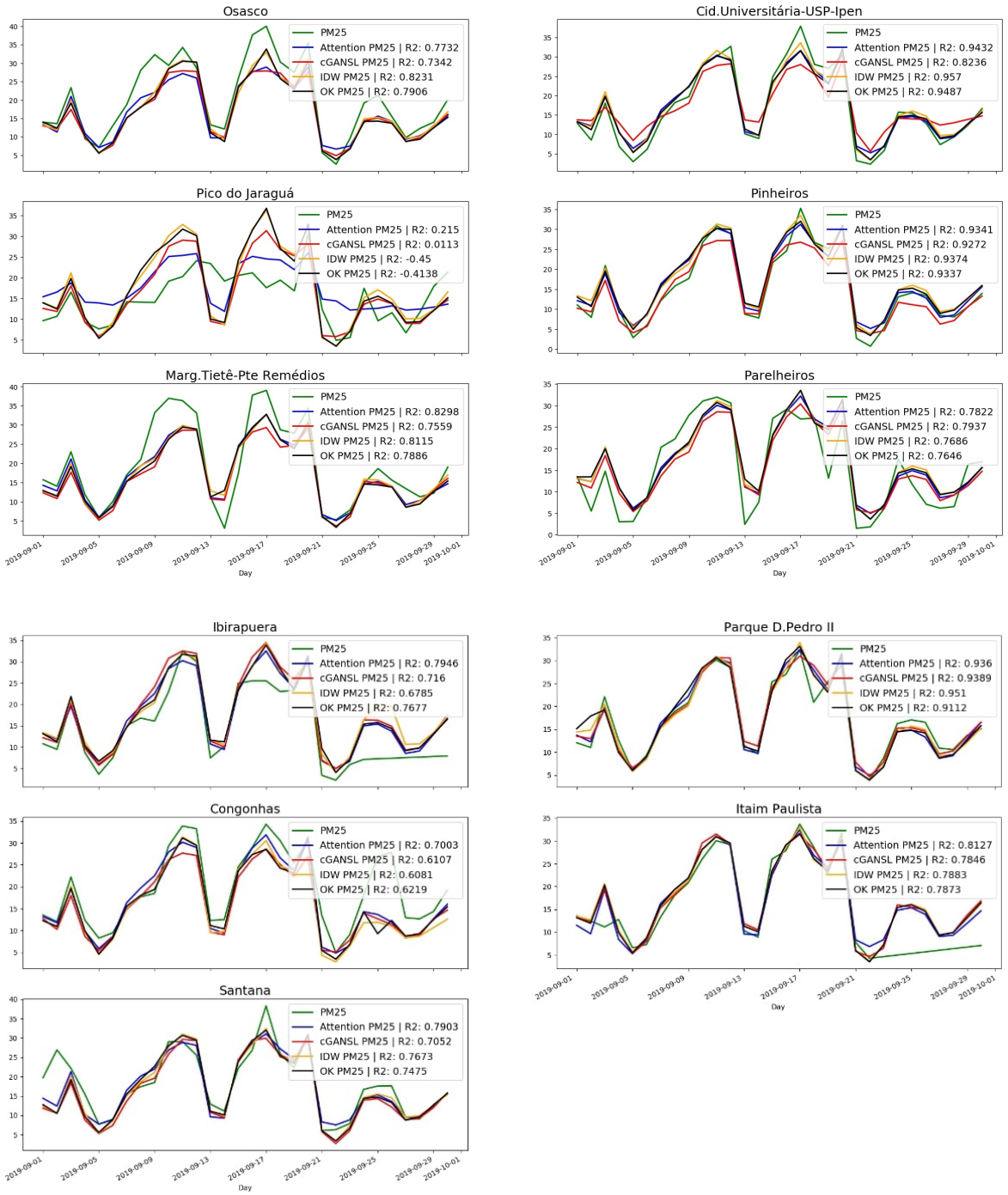


Figure 8.8: Outputs of spatial prediction models and observed values at air quality monitoring stations of São Paulo Network from 2019-09-01 to 2019-09-30. Source: Own.

We calculated the percentage of attention from the testing station (node i) to its nearest neighbors (nodes j), using 'Perceptron Affine' as the kernel in the attention layer, and the scaled Euclidean distance to its nearest neighbors shown in Figure 8.9. The testing node attends more to node 10 and less to nodes 8 and 9. It may be because

node 10 is the closest node according to euclidean distance. However, nodes 6 and 7 are more distant than node 3. Nonetheless, they have more attention weight than node 3. On the other hand, node 8 and node 9 have the same attention weight, but node 9 is closer than node 8 according to euclidean distance.

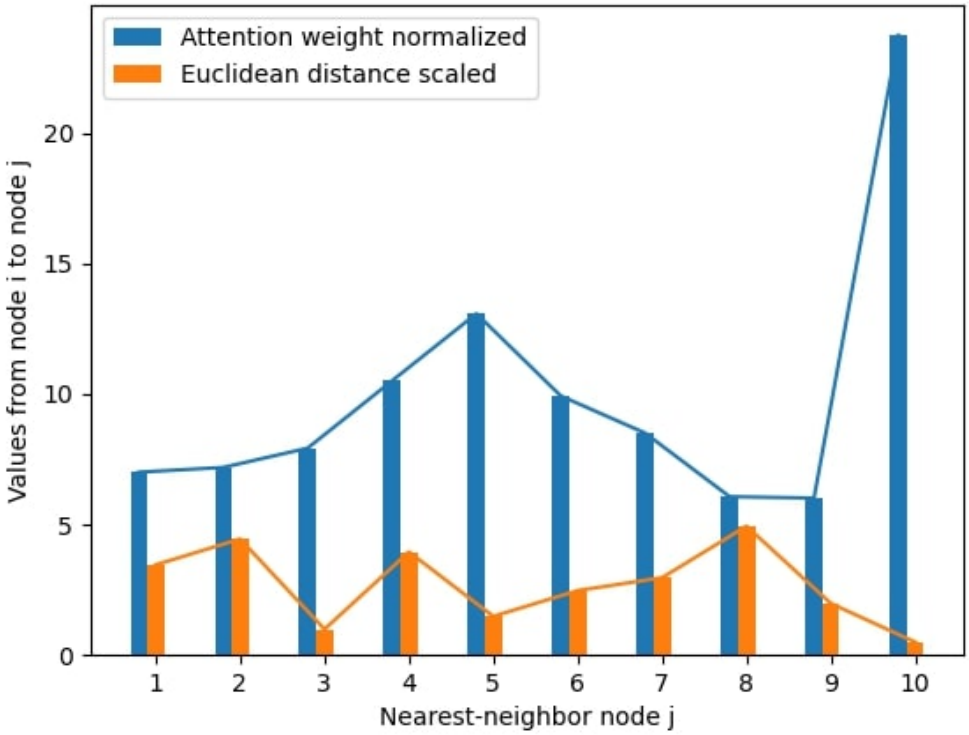


Figure 8.9: Normalized attention weights of testing station i to its 10 nearest-neighbors stations and the scaled spatial distance between node i and j. Source: Own

Besides, Figure 8.10 shows the location of four stations of São Paulo Network. Pico do Jauraguá is a location with high vegetation, while the nearest neighbor stations N.Senhora de O, Marg. Tiete Pte Remedios and Osasco. They are urban areas and have much less vegetation level. The spatial prediction models based on the spatial distance for selection of kNN (for example, IDW and cGANSL) have low performance predicting $PM_{2.5}$ in Pico do Jauraguá than the Neural Network with attention layer, shown in Figure 8.8. It may be because the attention layer does not directly consider the distance in a straight line to weight neighboring stations. The attention layer considers similar characteristics, such as the land type.

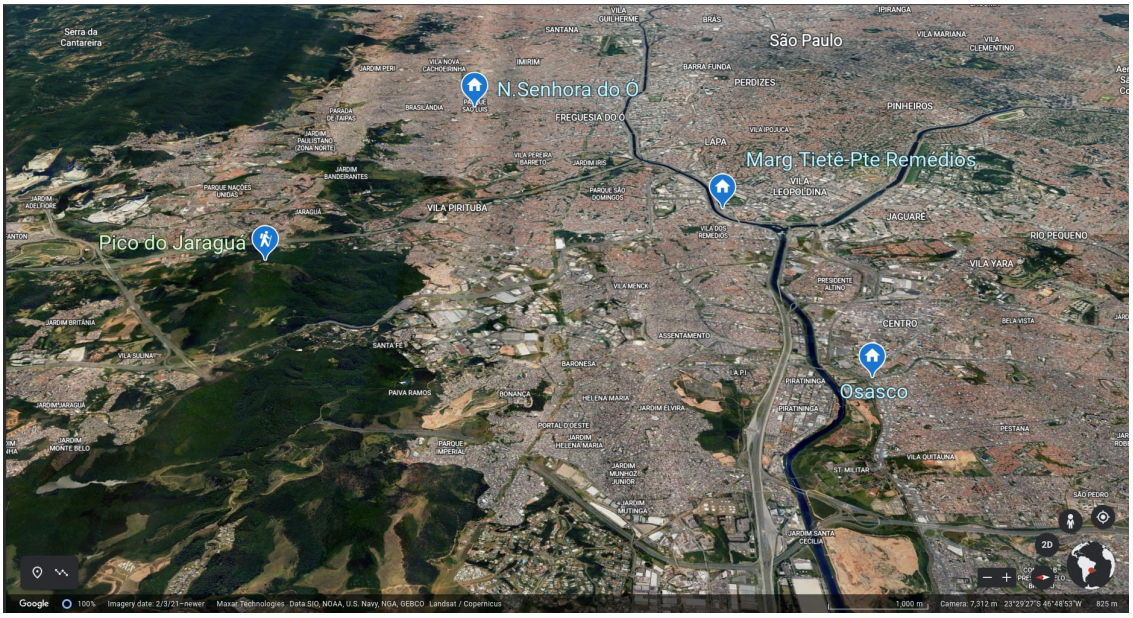


Figure 8.10: Station of Pico do Jauraguá and its three nearest neighbor stations. Source: Own

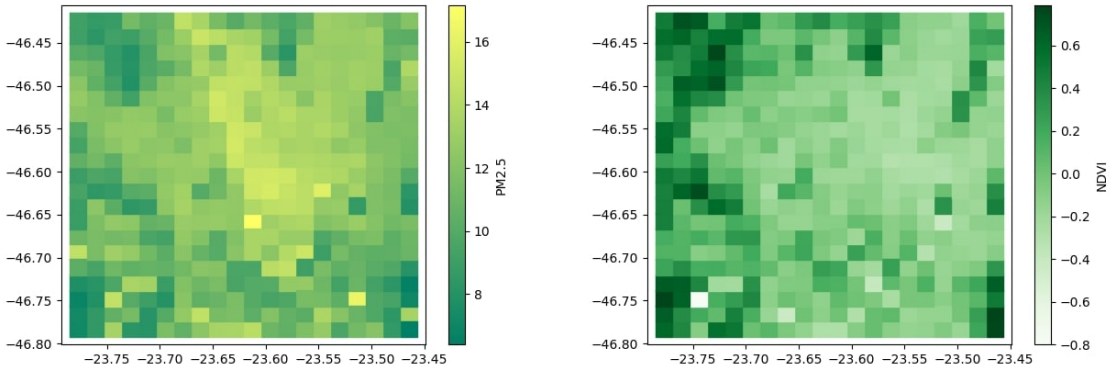
8.6 Land-related variables and predicted concentrations of fine particulate matter

Consequently, In Figure 8.11, we see that the pollution observed on January 3, 2017, and the predicted concentrations of $PM_{2.5}$ by the attention-based model, as well as the levels of NDVI are correlated. Most of the places in the center of the São Paulo region have low levels of vegetation, and these are areas with the highest levels of contamination. On the other hand, places with more vegetation have low pollution levels.

Moreover, Figure 8.12 shows a red box region, where the location with a higher DEM values has a higher level of pollution based on the $PM_{2.5}$ concentrations observed by AQM stations and predicted by our model for January 1, 2017.

8.7 Population-related variable and predicted concentrations of fine particulate matter

High concentrations of pollutants correlate with areas with a high level of night light (NTL), which indicates a high population and human activity based on what is shown by Figure 8.13. Meanwhile, darker locations indicate less population and uninhabitable regions, where at the same time pollution is less.



(a) (b)

Figure 8.11: (a) $PM_{2.5}$ concentrations observed by AQM stations and predicted by our model. (b) NDVI values were collected in each location of the bounding box region of São Paulo. Source: Own.

8.7.1 Region São Paulo Map of estimated Fine Particulate Matter by Spatial-Prediction Models

Figure 8.14 shows the heat maps of observation of $PM_{2.5}$ by AQM stations of Sao Paulo, predicted $PM_{2.5}$ concentrations on locations of São Paulo region, as well as vegetation and population levels infer from NDVI and NTL, respectively.

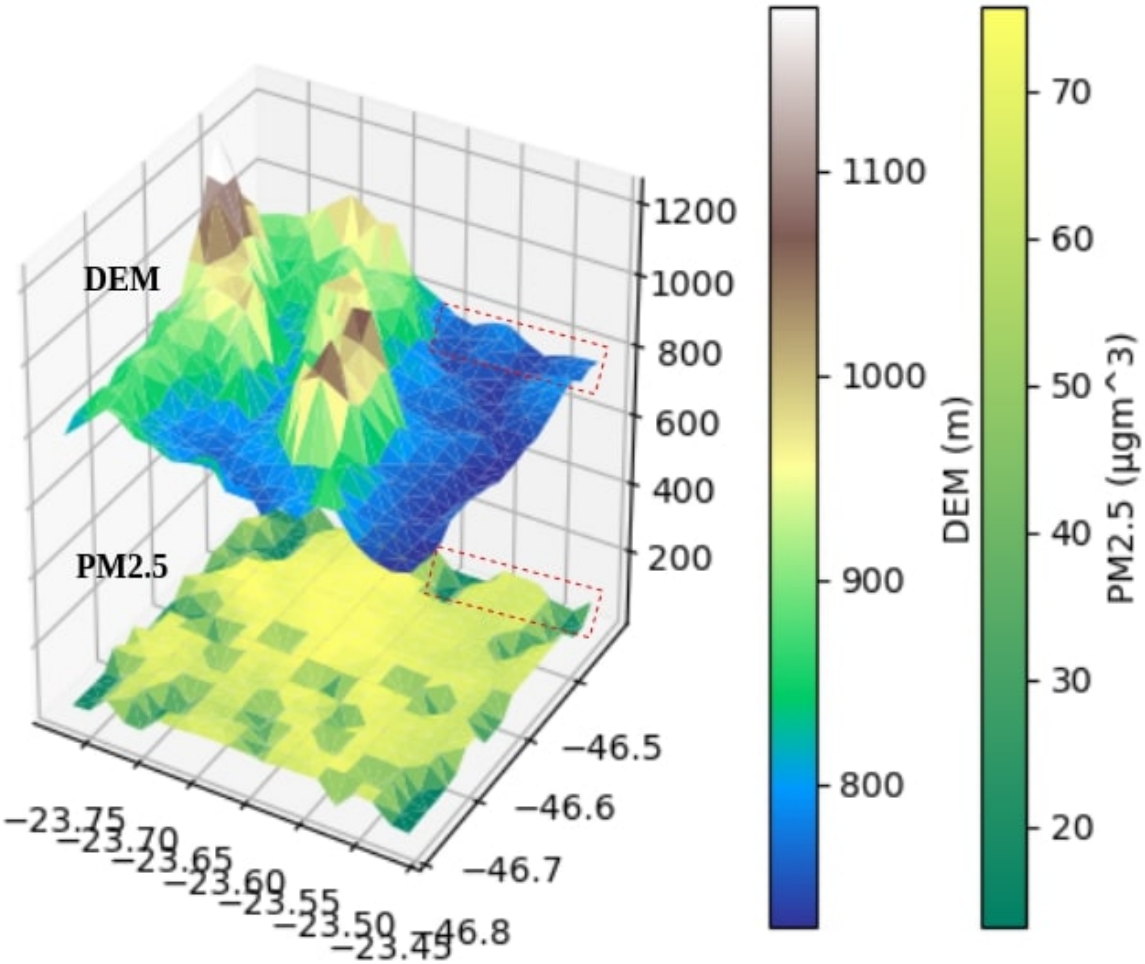
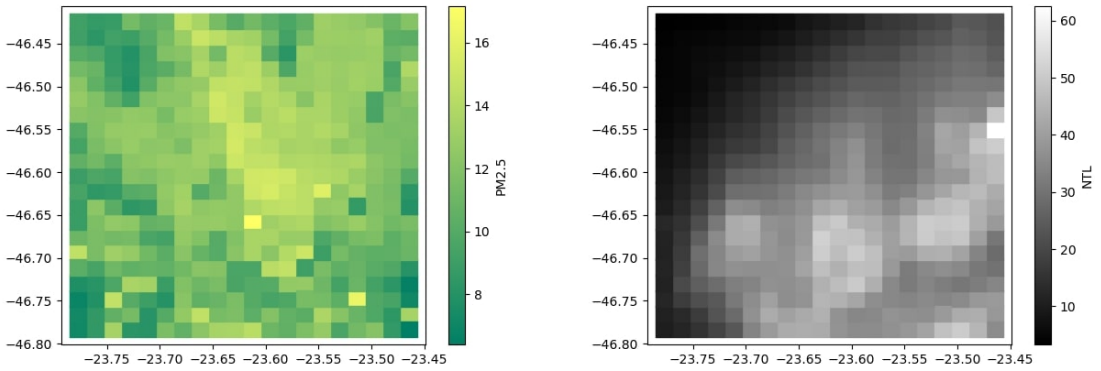


Figure 8.12: $PM_{2.5}$ concentrations and DEM values in the bounding box region of São Paulo.



(a) (b)

Figure 8.13: (a) $PM_{2.5}$ concentrations observed by AQM stations and predicted by our model. (b) NTL values were collected in each location of the bounding box region of São Paulo.

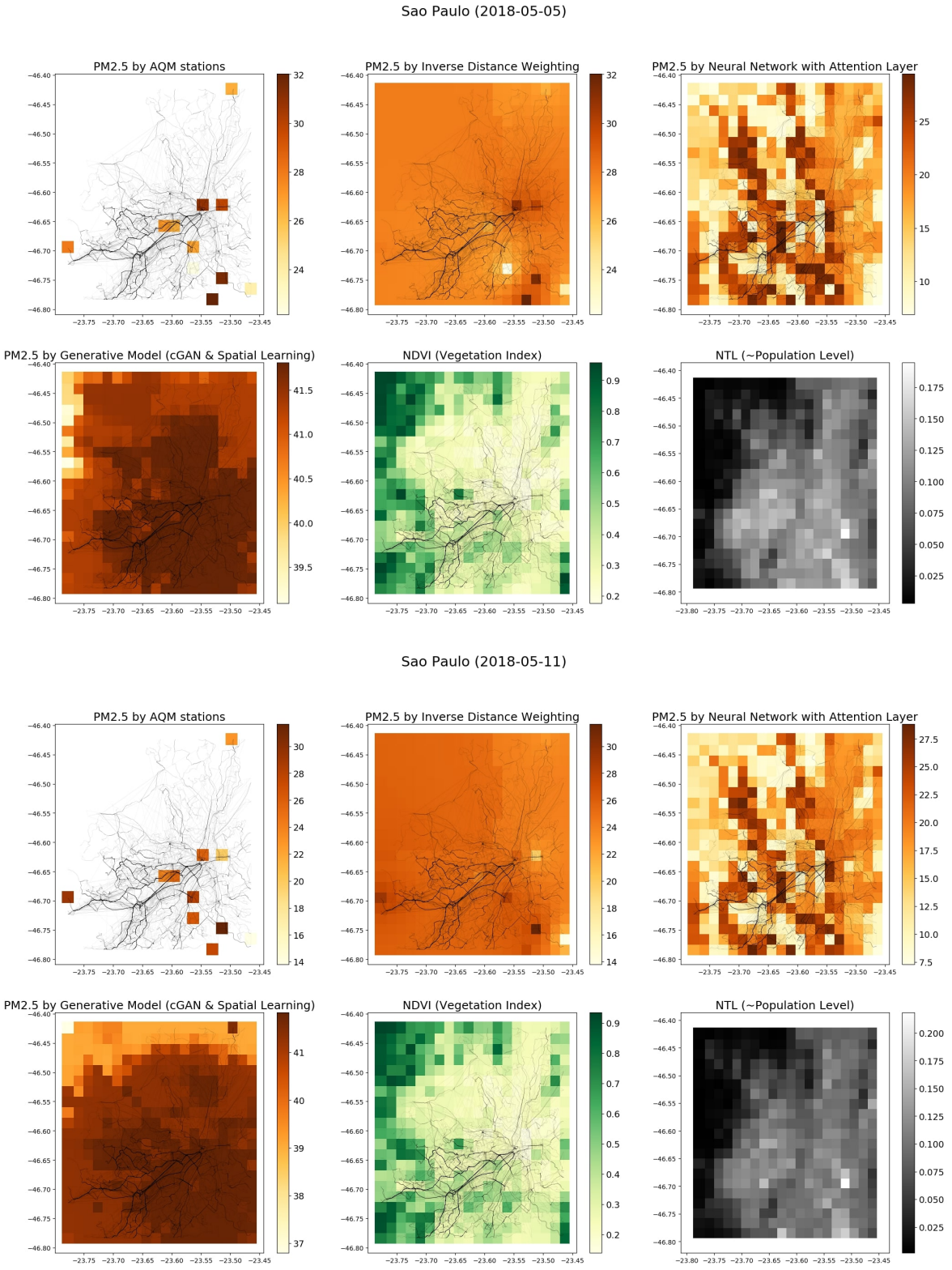


Figure 8.14: Heat maps of outputs of prediction models, observed concentrations of fine particulate, Normalized Difference Vegetation Index, and Nightime index, which infer the population level in some locations of São Paulo. Source: Own.

Chapter 9

Discussion and Conclusions

Prediction of air pollutant concentrations is a complicated process because spatial dispersion and temporal transition of pollutants depend on many variables. In the present study, we considered meteorological variables, variables related to land, and variables related to levels of human activity, as well as the observations of $PM_{2.5}$ of the kNN to infer $PM_{2.5}$ at a location without a measure of pollution.

We developed cGAN and cGANSL to infer $PM_{2.5}$. The closest neighbors in our proposed cGAN and cGANSL were selected by spatial distance. However, farther neighbors could have more relevant information than the more close. Therefore, we developed and experimented with the second method, which includes a kNN attention polling layer to quantify the attention that each node of the graph should give to its close neighbors, generating transformed feature representations for each node, which are entered to FCL2 for spatial prediction of $PM_{2.5}$ concentrations.

Moreover, we considered in this investigation two datasets for evaluating our models. The first is information from a Beijing region and the second from a São Paulo region. We found that the pollution data collected at the air quality monitoring stations of the Beijing network have a high correlation between them based on the analysis carried out. However, the São Paulo stations have little correlation between them.

The traditional interpolation models, especially the proposed cGANSL model that needs the selection of the kNN had a better performance to infer $PM_{2.5}$ in Beijing than the Neural Network with the attention-based layer. However, in São Paulo, where the stations are less correlated, the attention model performed better than all of them, based on RMSE, MAE error metrics, and R2 score.

In addition, we found that the "Pico do Jaraguá" station of the São Paulo network has a higher level of vegetation than its nearest neighbors. It is possible that the models based on the selection of kNN by Euclidean distance, such as IDW and cGANSL only consider these nearby stations due to their process of selection. However, distant stations could have the same features as Pico do Jaraguá and are not considered, thus

losing relevant information.

Therefore, the attention-based model considers all stations, weighting each station according to the level of attention it has to give each one, considering distant stations with similar features (similar vegetation levels or population levels). In this sense, the models based on the selection of kNN by spatial distance have lower performance than the attention model, mainly when testing them in predicting pollution in the location of Pico do Jaraguá.

Furthermore, the normalized attention weights calculated by the attention layer are not necessarily associated with the spatial distance, being able to give more attention weights to distant locations. Thus, the kNN attention polling layer is a key component in the model for improving the prediction of $PM_{2.5}$. Especially, the perceptron affine kernel based on GAT was the best in our experimentation.

Finally, we analyzed pollution concentrations predicted by Neural Network with an attention-based layer and NDVI values of São Paulo. We found that locations with more vegetation tend to have lower concentrations of $PM_{2.5}$ and areas with higher concentrations are related to less vegetation. Furthermore, high indices of NTL, a proxy of population density, are related to high $PM_{2.5}$ concentrations, indicating that regions with high human activity and more population are related to increased emission of pollutants. In contrast, we do not find a clear relationship between DEM and $PM_{2.5}$.

Chapter 10

Future Work

Future work is to add a module in the proposed models capable of learning temporal features to perform the temporal prediction of $PM_{2.5}$ on labeled locations and unlabeled locations predicted by our proposed attention model.

In addition, we are planning to perform transfer learning of our fine-tuned model to estimate fine particulate matter in other cities with similar weather and geographic factors to evaluate the generalization capacity of our model. Moreover, considering the purpose of implementing an air quality alert system, our model will be challenged to estimate other air pollutant varieties.

Bibliography

Agency, E. P. (2009). Integrated science assessment for particulate matter. Technical Report EPA/600/R-08/139F, U.S. Environmental Protection Agency.

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer, Cham.

Alqahtani, H., Kavakli-Thorne, M., et al. (2021). Applications of generative adversarial networks (gans): An updated review. *Archives of Computational Methods in Engineering*, 28(2):525–552.

Anh Nguyen, V., Starzyk, J., et al. (2012). Neural network structure for spatio-temporal long-term memory. *IEEE Transactions on Neural Networks*, 23:971–983.

Bahdanau, D., Cho, K., et al. (2015). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Braga, A. L., Conceição, G. M., et al. (1999). Air pollution and pediatric respiratory hospital admissions in são paulo, brazil. *Journal of Environmental Medicine*, 1(2):95–102.

Chen, J., Lu, J., et al. (2014). Seasonal modeling of pm2.5 in california's san joaquin valley. *Atmospheric Environment*, 92:182 – 190.

Chorowski, J., Bahdanau, D., et al. (2015). Attention-based models for speech recognition. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 577–585, Cambridge, MA, USA. MIT Press.

Chow, J. C. (1995). Measurement methods to determine compliance with ambient air quality standards for suspended particles. *Journal of the Air & Waste Management Association*, 45(5):320–382.

Colchado, L. E., Villanueva, E., et al. (2021). A neural network architecture with an attention-based layer for spatial prediction of fine particulate matter. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10.

Cressie, N. and Wikle, C. K. (2011). *Statistics for Spatio-Temporal Data*. Wiley.

DAAC, L. Viirs/npp daily gridded day night band 500m linear lat lon grid night - laads daac. https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/VNP46A1. (Accessed on 06/04/2021).

de Fatima Andrade, M., Kumar, P., et al. (2017). Air quality in the megacity of são paulo: Evolution over the last 30 years and future perspectives. *Atmospheric Environment*, 159:66–82.

Delavar, M. R., Gholami, A., et al. (2019). A novel method for improving air pollution prediction based on machine learning approaches: A case study applied to the capital city of tehran. *ISPRS International Journal of Geo-Information*, 8(2).

Di, Q., Amini, H., et al. (2019). An ensemble-based model of pm2.5 concentration across the contiguous united states with high spatiotemporal resolution. *Environment International*, 130:104909.

Didan, K. (2015). Mod13a2 MODIS/terra Vegetation Indices 16-Day L3 Global 1km SIN Grid V006 [Data set]. *NASA EOSDIS Land Processes DAAC. Accessed*, 2021-06-04.

Du, Y., Xu, X., et al. (2015). Air particulate matter and cardiovascular disease: the epidemiological, biomedical and clinical evidence. *Journal of Thoracic Disease*, 8(1).

Fan, J., Li, Q., et al. (2017). A spatiotemporal prediction framework for air pollution based on deep rnn. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W2:15–22.

Farr, T. G., Rosen, P. A., et al. (2007). The shuttle radar topography mission. *Reviews of Geophysics*, 45(2).

Foley, K. M., Roselle, S. J., et al. (2010). Incremental testing of the community multiscale air quality (cmaq) modeling system version 4.7. *Geoscientific Model Development*, 3(1):205–226.

Friedjungová, M., Vasata, D., et al. (2020). Missing features reconstruction using a wasserstein generative adversarial imputation network. *CoRR*, abs/2006.11783.

Gao, Y., Liu, L., et al. (2020). SI-AGAN: spatial interpolation with attentional generative adversarial networks for environment monitoring. In Giacomo, G. D., Catalá, A., et al., editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 1786–1793. IOS Press.

Goodfellow, I., Bengio, Y., et al. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goodfellow, I. J., Pouget-Abadie, J., et al. (2014). Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA. MIT Press.

Guan, W.-J., Zheng, X.-Y., et al. (2016). Impact of air pollution on the burden of chronic respiratory diseases in china: time for urgent action. *The Lancet*, 388(10054):1939–1951.

Gui, J., Sun, Z., et al. (2020). A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937*.

Guo, C., Liu, G., et al. (2020). An unsupervised pm2.5 estimation method with different spatio-temporal resolutions based on kidw-tcgru. *IEEE Access*, 8:190263–190276.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Iskandaryan, D., Ramos, F., et al. (2023). Graph neural network for air quality prediction: A case study in madrid. *IEEE Access*, 11:2729–2742.

Isola, P., Zhu, J., et al. (2017). Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976.

Jabbar, A., Li, X., et al. (2021). A survey on generative adversarial networks: Variants, applications, and training. *ACM Comput. Surv.*, 54(8).

Kalb, V., Román, M., et al. (n.d). NASA viirs land.

Kumar, A. and Goyal, P. (2011). Forecasting of daily air quality index in delhi. *Science of The Total Environment*, 409(24):5517 – 5523.

Li, D., Yu, H., et al. (2021). Ddgnet: A dual-stage dynamic spatio-temporal graph network for pm2.5 forecasting. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1679–1685.

Li, L., Losser, T., et al. (2014). Fast inverse distance weighting-based spatiotemporal interpolation: A web-based application of interpolating daily fine particulate matter pm2.5 in the contiguous u.s. using parallel programming and k-d tree. *International Journal of Environmental Research and Public Health*, 11(9):9101–9141.

Li, S., Xie, G., et al. (2020). Urban pm2.5 concentration prediction via attention-based cnn–lstm. *Applied Sciences*, 10(6).

Li, T., Shen, H., et al. (2017a). Estimating ground-level pm2.5 by fusing satellite and station observations: A geo-intelligent deep learning approach. *Geophysical Research Letters*, 44(23):11,985–11,993.

Li, X., Peng, L., et al. (2016). Deep learning architecture for air quality predictions. *Environmental Science and Pollution Research*, 23:22408–22417.

Li, X., Peng, L., et al. (2017b). Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental Pollution*, 231:997 – 1004.

Longley, P. (2005). *Geographical information systems : principles, techniques, management and applications*. John Wiley, Hoboken (New Jersey), 2nd ed., abridged. edition.

Lowsen, D. H. and Conway, G. A. (2016). Air pollution in major chinese cities: Some progress, but much more to do. *Journal of environmental protection*, 7(13):2081–2094.

Ma, J., Ding, Y., et al. (2019). Spatiotemporal prediction of pm2.5 concentrations at different time granularities using idw-blstm. *IEEE Access*, 7:107897–107907.

Ma, L., Gao, Y., et al. (2017). Estimation of ground pm2.5 concentrations using a dem-assisted information diffusion algorithm: A case study in china. *Scientific Reports*, 7(1):15556.

Ma, T. and Zhang, A. (2019). Affinitynet: Semi-supervised few-shot learning for disease type prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1069–1076.

Maas, A. L., Hannun, A. Y., et al. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer.

Masroor, K., Fanaei, F., et al. (2020). Spatial modelling of pm2.5 concentrations in tehran using kriging and inverse distance weighting (idw) methods. *Journal of Air Pollution and Health*, 5(2):89–96.

Mintz, D., . U. S. (2009). *Technical assistance document for the reporting of daily air quality–the Air Quality Index (AQI) [electronic resource].* U.S. Environmental Protection Agency, Office of Air Quality Planning and Standards Research Triangle Park, N.C.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *CoRR*, abs/1411.1784.

Najjar, Y. (2011). Gaseous pollutants formation and their harmful effects on health and environment. *Ashdin Publishing Innovative Energy Policies*, 1.

Nieto, P. G., Combarro, E., et al. (2013). A svm-based regression model to study the air quality at local scale in oviedo urban area (northern spain): A case study. *Applied Mathematics and Computation*, 219(17):8923 – 8937.

Noda, L., Nóbrega, A. B. E. Q., et al. (2021). Covid-19: Has social isolation reduced the emission of pollutants in the megacity of são paulo—brazil? *Environment, Development and Sustainability*, 23(8):12233–12251.

Organization, W. H. (2013). Health effects of particulate matter. policy implications for countries in eastern europe, caucasus and central asia (2013).

Osowski, S. and Garanty, K. (2007). Forecasting of the daily meteorological pollution using wavelets and support vector machine. *Engineering Applications of Artificial Intelligence*, 20(6):745 – 755.

Paschalidou, A. K., Karakitsios, S., et al. (2011). Forecasting hourly pm10 concentration in cyprus through artificial neural networks and multiple regression models: implications to local environmental management. *Environmental Science and Pollution Research*, 18(2):316–327.

Pražnikar, Z. and Pražnikar, J. (01 Jan. 2012). The effects of particulate matter air pollution on respiratory health and on the cardiovascular system. *Slovenian Journal of Public Health*, 51(3):190 – 199.

Pruthi, D. and Liu, Y. (2022). Low-cost nature-inspired deep learning system for pm2.5 forecast over delhi, india. *Environment International*, 166:107373.

Qi, Z., Wang, T., et al. (2018). Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2285–2297.

Ramachandran, P., Parmar, N., et al. (2019). Stand-alone self-attention in vision models.

Reátegui-Romero, W., Sánchez-Ccoyllo, O. R., et al. (2018). Pm2.5 estimation with the wrf/chem model, produced by vehicular flow in the lima metropolitan area. *Open Journal of Air Pollution*, page 215–243.

Rumelhart, D. E., Hinton, G. E., et al. (1986). Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536.

Saide, P. E., Carmichael, G. R., et al. (2011). Forecasting urban pm10 and pm2.5 pollution episodes in very stable nocturnal conditions and complex terrain using wrf–chem co tracer model. *Atmospheric Environment*, 45(16):2769 – 2780.

Sánchez-Ccoyllo, O. R., Ordoñez-Aquino, C. G., et al. (2018). Modeling study of the particulate matter in lima with the wrf-chem model: Case study of april 2016. *International journal of applied engineering research : IJAER*, 13(11):10129–10141.

Sayer, A. M., Hsu, N. C., et al. (2013). Validation and uncertainty estimates for modis collection 6 "deep blue" aerosol data. *Journal of Geophysical Research: Atmospheres*, 118(14):7864–7872.

Schloeder, C., Zimmerman, N., et al. (2001). Comparison of methods for interpolating soil properties using limited data. *Soil Science Society of America Journal*, 65(2):470–479.

Shi, P., Fang, X., et al. (2021). An improved attention-based integrated deep neural network for pm2.5 concentration prediction. *Applied Sciences*, 11(9).

Shin, M., Kang, Y., et al. (2020). Estimating ground-level particulate matter concentrations using satellite-based data: a review. *GIScience & Remote Sensing*, 57(2):174–189.

Taborda, R., Datin, N., et al. (2020). Exploring air quality using a multiple spatial resolution dashboard — a case study in lisbon. In *2020 24th International Conference Information Visualisation (IV)*, pages 140–145.

Tang, X., Yao, H., et al. (2020). Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. In *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, AAAI 2020 - 34th AAAI Conference

on Artificial Intelligence, pages 5956–5963. AAAI press. Funding Information: This material is based upon work supported by, or in part by, the National Science Foundation (NSF) under grant 1909702. Publisher Copyright: © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.; 34th AAAI Conference on Artificial Intelligence, AAAI 2020 ; Conference date: 07-02-2020 Through 12-02-2020.

Toutouh, J. (2021). Conditional generative adversarial networks to model urban outdoor air pollution. In Nesmachnow, S. and Hernández Callejo, L., editors, *Smart Cities*, pages 90–105, Cham. Springer International Publishing.

Vargas-Campos, I. R. and Villanueva, E. (2021). Comparative study of spatial prediction models for estimating pm$$_{2.5}$$concentration level in urban areas. In Lossio-Ventura, J. A., Valverde-Rebaza, J. C., et al., editors, *Information Management and Big Data*, pages 169–180, Cham. Springer International Publishing.

Veličković, P., Cucurull, G., et al. (2018). Graph attention networks.

Wang, W. and Guo, Y. (2009). Air pollution pm2.5 data analysis in los angeles long beach with seasonal arima model. In *2009 International Conference on Energy and Environment Technology*, volume 3, pages 7–10.

Wang, W., Zhao, S., et al. (2019). Estimation of pm2.5 concentrations in china using a spatial back propagation neural network. *Scientific Reports*, 9(1):13788.

Webster, R., Webster, R. R., et al. (2007). *Geostatistics for environmental scientists / Richard Webster and Margaret A. Oliver.* Statistics in practice. Wiley, Chichester, 2nd ed. edition.

Wei, J., Huang, W., et al. (2019). Estimating 1-km-resolution pm2.5 concentrations across china using the space-time random forest approach. *Remote Sensing of Environment*, 231:111221.

Wikle, C. K., Zammit-Mangion, A., et al. (2019). *Spatio-Temporal Statistics with R (Chapman & Hall/CRC The R Series).* Chapman and Hall/CRC.

Wong, D. W., Yuan, L., et al. (2004). Comparison of spatial interpolation methods for the estimation of air quality data. *Journal of Exposure Science & Environmental Epidemiology*, 14(5):404–415.

Yang, Q., Yuan, Q., et al. (2019). The relationships between pm2.5 and aerosol optical depth (aod) in mainland china: About and behind the spatio-temporal variations. *Environmental Pollution*, 248:526 – 535.

Zhang, G., Rui, X., et al. (2018a). Critical review of methods to estimate pm2.5 concentrations within specified research region. *ISPRS International Journal of Geo-Information*, 7(9).

Zhang, L., Lin, J., et al. (2018b). Trend analysis and forecast of pm2.5 in fuzhou, china using the arima model. *Ecological Indicators*, 95:702 – 710.

Zheng, Y., Liu, F., et al. (2013). U-air: When urban air quality inference meets big data. In *Proceedings of the 19th SIGKDD conference on Knowledge Discovery and Data Mining (KDD 2013)*.

Zou, B., Wang, M., et al. (2015). Spatial modeling of pm2.5 concentrations with a multifactoral radial basis function neural network. *Environmental Science and Pollution Research*, 22.