MACHINE LEARNING APPROACHES TO PREDICT $PM_{2.5}$ USING SATELLITE IMAGES

by
Xinyu Du

A thesis submitted to Johns Hopkins University in conformity with the requirements for the degree of Master of Science

Baltimore, Maryland
May 2022

© 2022 Xinyu Du
All rights reserved

Abstract

According to World Health Organization, air pollution is considered to be one of the greatest environmental health threats. $PM_{2.5}$, fine particles with a diameter that is generally 2.5 micrometers and smaller, is inhalable into the lungs and can induce adverse health effects. In order to mitigate the effects of $PM_{2.5}$ on health outcomes, it is crucial to make accurate predictions of ambient concentrations. In the past, most studies developed traditional linear models from meteorological data or Environmental Protection Agency (EPA) ambient air quality monitors. However, the non-linear relationship between $PM_{2.5}$ and other factors impacts the effectiveness of the models. Some other barriers are the sparseness of air quality monitors and the limited data sources, which also hinder researchers' ability to get accurate predictions. Recently, advanced technologies in satellite remote sensing have been widely used to estimate $PM_{2.5}$ concentrations. The implementation of machine learning approaches has improved computational efficiency and accuracy.

In this study, we used daily satellite imagery from January 2017 to October 2021 in 25 U.S. locations, and implemented an eXtreme Gradient Boosting (XGBoost) algorithm, a deep Convolutional Neural Network (CNN), and a CNN-XGBoost pipeline to make predictions based on the extracted features from each satellite image and atmospheric information along with meteorological conditions. To evaluate the performance of each model, we used daily EPA Federal Reference Method $PM_{2.5}$ measurements as the validation data, and calculated the corresponding root mean squared error (RMSE) and the coefficient of determinant (R^2). After combining each daily observation from

25 locations, the XGBoost approach demonstrated the highest performance with an RMSE of 3.98 μg m^{-3} and an R^2 of 0.65. The CNN-XGBoost pipeline, tending to overestimate $PM_{2.5}$ concentrations, had an RMSE of 5.87 μg m^{-3} and an R^2 of 0.37. In conclusion, our study showed that XGBoost achieved reasonable $PM_{2.5}$ prediction performance, indicating that the application of satellite remote sensing data and machine learning approaches has significant potential use in $PM_{2.5}$ concentrations prediction.

The data and R code used in this thesis is available on GitHub (https://github.com/sindydu0904/Satellite_pmPredict).

Advisor: Dr. Roger D. Peng

Secondary Reader: Dr. Abhirup Datta

Acknowledgments

First and most, I would like to express my sincere gratitude to my advisor, Dr. Roger D. Peng for his continuous guidance and support from the start of this program. His enthusiasm, patience, motivation, and immense knowledge have deeply inspired me. Without his persistent help, this thesis would not have been possible.

I am deeply grateful to Dr. Abhirup Datta who is the second reader of this thesis, for his insightful comments and suggestions. I would like to thank Dr. Elizabeth Colantuoni and Mary Joy Argo for their consistent help throughout my time at Hopkins.

Last but not the least, I would like to thank my parents and friends for their unfailing love and encouragement. To my best friend Yutong Li, thanks for always being there for me especially during the hard times.

Contents

Abstract ii

Acknowledgments iv

Contents v

List of Tables vii

List of Figures viii

Chapter 1 Introduction 1

Chapter 2 Data 6

2.1 Satellite Imagery 6

2.2 EPA Data 8

2.3 Meteorological Measurements 11

Chapter 3 Methods 14

3.1 Extreme Gradient Boosting for All Locations 14

3.2 Deep Convolution Neural Network for Each Location 17

3.3 Comparison and Combination 19

3.3.1 Comparison: two models for Each Location 19

3.3.2 Combination: CNN-XGBoost Pipeline for All Locations 20
3.4 Evaluation Criteria 20

Chapter 4 Results **22**
4.1 XGBoost 22
4.2 Cloud Cover Issue 24
4.3 Comparison of Two Models 27
4.4 CNN-XGBoost Pipeline 28

Chapter 5 Discussion **35**
5.1 Comparison with Related Studies 35
5.2 Limitations and Future Work 38

Chapter 6 Conclusion **39**

References **40**

List of Tables

2-1 Basic and Atmospheric Information of 25 Locations 10

2-2 Meteorological Conditions of 25 Locations 12

4-1 XGBoost RMSE by Month of Testing Set 23

4-2 XGBoost RMSE by Year of Testing Set 24

4-3 XGBoost Accounting for Cloud Cover Issue by Location 26

4-4 XGBoost Accounting for Cloud Cover Issue by Month 27

4-5 Model Comparison for Each Location 29

5-1 Comparison of Model Performance with Other Studies 35

List of Figures

2-1 Physical Locations of 25 Places 7

2-2 Illustration of PlanetScope satellites rotating around the Earth 8

2-3 Satellite Imaginary from Akron and Austin 9

2-4 Time Series of $PM_{2.5}$ Concentrations for Akron and Austin . 13

3-1 Illustration of Splitting an Image 16

4-1 Scatter Plot of XGboost Predicted v.s. EPA 25

4-2 25 Time Series Plots of CNN Predicted v.s. EPA 30

4-3 25 Time Series Plots of XGBoost Predicted v.s. EPA 31

4-4 25 Scatter Plots of CNN Predicted v.s. EPA 32

4-5 25 Scatter Plots of XGboost Predicted v.s. EPA 33

4-6 Scatter Plot of CNN-XGboost Predicted v.s. EPA 34

Chapter 1

Introduction

Air pollution is a mixture of gaseous pollutants and particle matter (PM), each of which has deleterious effects on human health. In 2019, air pollution was the 4th leading risk factor for death globally, and over 99% of the world's population was living in areas where air quality levels exceeded World Health Organization limits (Murray et al., 2020). $PM_{2.5}$, fine inhalable particles with a diameter that is generally 2.5 micrometers and smaller, is one of the most dominant contributors to air pollution. It poses a considerable risk to human health because it can penetrate deeply into the lungs and may cause chronic asthma (Keet, Keller, and Peng, 2018; Fan et al., 2016), respiratory inflammation (Xing et al., 2016; Hooper et al., 2018), cardiovascular diseases (Lipsett et al., 2011; Hamanaka and Mutlu, 2018; Hayes et al., 2020), premature deaths (O'Donnell et al., 2011, Di, Dai, et al., 2017), diabetes (Manisalidis et al., 2020), and mental disorders (Roberts et al., 2019). Moreover, recent study reports that wildfires could amplify the effect of short-term exposure to $PM_{2.5}$ on COVID-19 cases and deaths (X. Zhou et al., 2021).

To mitigate the adverse health effects of $PM_{2.5}$, Environmental Protection Agency (EPA) regulates inhalable particles by setting National Ambient Air Quality Standards (NAAQS) and implementing the Air Quality System (AQS). AQS is the EPA repository of ambient air quality data. The $PM_{2.5}$ monitoring network is comprised of manual Federal Reference Methods (FRMs) or automated continuous Federal Equivalent

Methods (FEMs), which are used to assess compliance with NAAQS. This network contains sites operated by State, local government, and monitoring agencies. Currently there are 1,381 active $PM_{2.5}$ AQS stations in the U.S. that can provide daily or annual summary data. Most epidemiological studies rely on these monitors to assess the influence of $PM_{2.5}$ on health outcomes (Z. Li et al., 2019).

However, these monitors are relatively sparse in certain regions of the country and may be far away from the pollution sources due to poor infrastructure and high costs of their installation, operation, and maintenance (Chow, 1995). States like Idaho, Kansas, and Rhode Island have fewer than 10 monitors. In some relatively large cities, such as Baltimore (MD), New Haven (CT) and Syracuse (NY), there is only one monitor. But there can be dramatic differences in $PM_{2.5}$ concentrations within the states. In fact, $PM_{2.5}$ concentrations within counties or cities may vary significantly from one neighborhood to another (Borghi et al., 2021). The sparseness of AQS stations has hindered scientists' ability to detect the high spatial variability of $PM_{2.5}$, and limited the precision of $PM_{2.5}$ exposure assessment.

The rise of microsensor technology is contributing to the broad application of low-cost sensors for air quality monitoring. These low-cost sensors can be used for supplementing existing monitoring network, and thus increase the density of AQS stations by providing real time measurements at much lower cost while allowing higher spatial coverage (Kumar et al., 2015). Another advantage of these low-cost sensors is the low energy consumption, as the power supply voltage is around 5 V and the working current is usually lower than 250 mA (Badura et al., 2018). In addition, they can be installed easily and operated without human intervention (Morawska et al., 2018), making it possible for users to monitor air pollution without any technical knowledge.

However, low-cost sensors have their drawbacks. The accuracy and quality of the collected data turns into a major concern, which limits the widespread adoption of

low-cost sensor technology. Unreliable data may lead to detrimental consequences, such as misreporting or mispredicting the air pollutant levels when they are above the thresholds. Since these sensors are built without prior modifications and are not designed according to standardized procedures, they require calibration before deployment (Piedrahita et al., 2014, Gao, J. Cao, and Seto, 2015).

Remote sensing offers a solution for overcoming the limitations of low-cost sensors. It is the acquisition of information about an object (without being in physical contact with it) through measuring its reflected and emitted radiation from satellite. Over the last decade, due to the relatively inexpensive cost of launching satellites and the employment of low-orbit satellite constellations, the use of satellite remote sensing has greatly increased. Satellite remote sensing is capable of observing large regions of the Earth and allows for collecting high-resolution imagery, which leads to a great reduction in cost and resource utilization. Early work used satellite remote sensing data for land cover classification (Ozesmi and Bauer, 2002), archaeological fieldwork projects (Parcak, 2009), monitoring wetland resources (Ozesmi and Bauer, 2002), oil spill detection (Brekke and Solberg, 2005) and forestry planning (Holmgren and Thuresson, 1998).

In recent years, scientists applied satellite remote sensing technology to study climate change (Yang et al., 2013), surface air quality (Martin, 2008), and to estimate $PM_{2.5}$ (Lin et al., 2015, Y. Liu et al., 2005, Van Donkelaar, Martin, and R. J. Park, 2006). Most studies used measurements of aerosol optical depth (AOD) or Moderate Resolution Imaging Spectroradiometer (MODIS) as predictors, and applied linear models to investigate the relationship between them and $PM_{2.5}$. But the lack of accurate information on particle size distribution and composition leads to inaccurate predictions, suggesting a substantial amount of variability in $PM_{2.5}$ concentrations cannot be explained by those models (Y. Liu et al., 2005).

With the rapid growth in the availability of satellite remote sensing data and

advanced computational technologies, multiple machine learning frameworks have been proposed for forecasting air pollution (Yanosky et al., 2014, Holloway and Mengersen, 2018). Recent studies utilized some common non-linear models, as well as machine learning algorithms like Random Forest (Zamani Joharestani et al., 2019, Sun, Gong, and J. Zhou, 2021, Guo et al., 2021), Extreme Gradient Boost (Ma et al., 2020, Just et al., 2018), and Deep Learning methods (Zamani Joharestani et al., 2019, T. Zheng et al., 2020, Muthukumar et al., 2021). These machine learning methods can be used to predict $PM_{2.5}$ concentrations in certain areas where traditional monitoring networks are not available.

Although these machine learning approaches have demonstrated their feasibility and accuracy in addressing nonlinear characteristics when using satellite images to make predictions, most of these models have not been tested in multiple states across the U.S.. In this study, we implemented and evaluated following two methods to forecast daily $PM_{2.5}$ concentrations of 25 locations in 17 states across the U.S..

Extreme Gradient Boost (XGBoost), developed as a research project by Tianqi Chen and Carlos Guestrin in 2016, is a machine learning technique that can be used for both classification and regression problems (T. Chen and Guestrin, 2016). It is a performant and efficient algorithm that can handle large datasets (T. Chen, T. He, et al., 2015). XGBoost consists of a loss function, usually a linear model for regression, and a regularization term. It generates a weak learner at each step, accumulating better predictions while discarding worse ones to form the final good predictions.

Convolution Neural Network (CNN) is an architecture consisting of several convolutional layers and pooling layers, followed by a couple of fully-connected layers, and a fully-connected regression layer with linear or sigmoid activation (O'Shea and Nash, 2015). It is a powerful algorithm for image processing. Each color image is stored in a 3-dimensional array to be fitted in the model, where the first two dimensions correspond to the height and width of the image (the number of pixels) while the last

dimension corresponds to the red, green, blue and near-infrared colors present in each pixel.

Due to complexity of the weather and variability of the seasons, meteorological measurements (temperature, dew point temperature, relatively humidity, surface pressure, precipitation and wind speed) and atmospheric data (AOD, cloud cover ratio, ground sampling distance of the image acquisition, water vapor concentration used, and ozone concentration used) have been identified as important factors in $PM_{2.5}$ concentrations prediction (Stowell et al., 2020, W. Liu et al., 2019, Kleine Deters et al., 2017).

To further enhance the estimation accuracy, this thesis aims to 1) combine all observations from all locations and use an XGBoost approach to model the association between $PM_{2.5}$ and atmospheric data along with meteorological information, 2) compare two models' performances in each location using only the extracted features from satellite imagery, and 3) propose a CNN-XGBoost pipeline, where a CNN is used to process the extracted features that characterize the dynamic changes among the satellite imagery, and an XGBoost is used to model the association between $PM_{2.5}$ and predictions from CNN, atmospheric data along with meteorological information.

Chapter 2

Data

Satellite imagery data that we could obtain were limited by the PlanetScope Application Programming Interface (API). Among the locations with available satellite imagery, we chose some of them for this study based on the corresponding 1) availability of daily data collected from EPA monitors, and 2) wide geographic distribution across the U.S. As a result, 25 locations across the U.S. were selected because both $PM_{2.5}$ measurements from EPA monitors and daily PlanetScope satellite images from surrounding area were abundant from 2017 to 2021. Figure 2-1 shows the physical location of each place.

2.1 Satellite Imagery

A satellite constellation is a group of similar satellites, which can be used for navigation, Earth observation, and satellite telephony. PlanetScope, operated by Planet (https://www.planet.com/), is one of the satellite constellations that produces images of land surface of the Earth. The PlanetScope satellite constellation consists of more than 200 individual satellites called "Doves". Each Dove satellite is made of a 10 cm x 10 cm x 30 cm CubeSat 3U form factor. Unlike a single satellite which only covers a small area everyday, the PlanetScope satellite constellation is able to provide an image of most Earth's landmass at 3-meter resolution per day by making all active

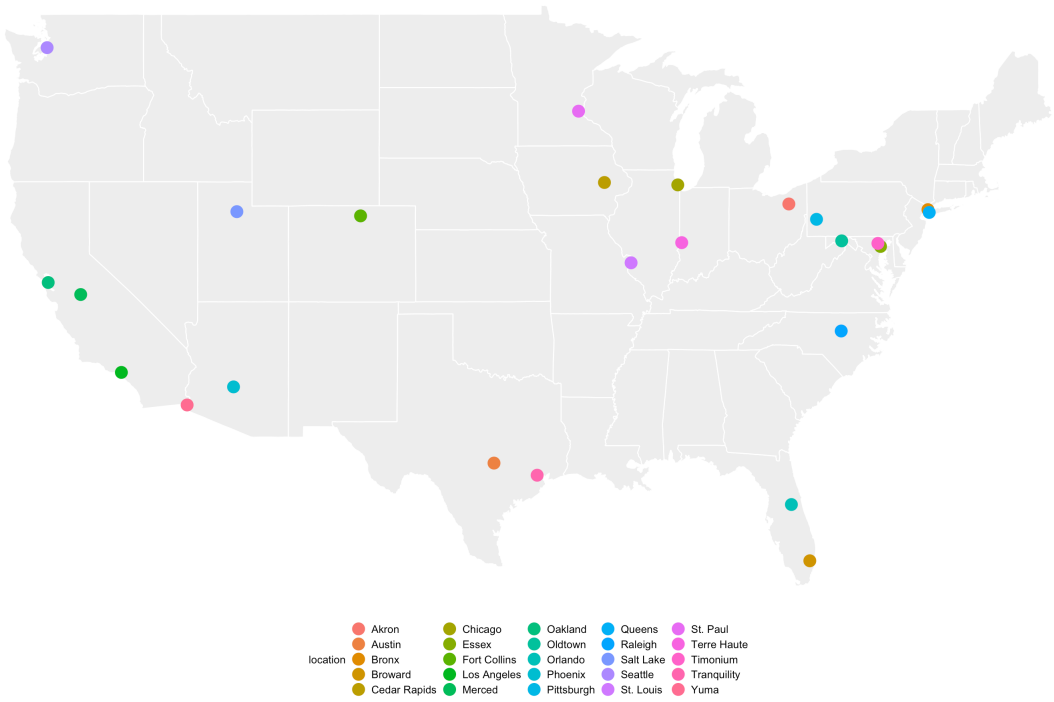


Figure 2-1: Physical Locations of 25 Places

satellites orbit Earth every 90 minutes. Thus, at least one satellite is visible anytime and anywhere on Earth. Each image is captured as a continuous strip of single frame images known as "scenes". Scenes are acquired as 4 multispectral bands (*i.e.* blue, green, red and near-infrared). PlanetScope collects up to 350 million square kilometers of imagery every day to create a massive archive of global satellite data. Figure 2-2 shows how PlanetScope satellites, represented by the white circles, continuously rotate around the Earth. Figure 2-3 (a) and (b) show satellite images in Akron (OH) on January 6, 2017 and July 3, 2017. Figure 2-3 (c) and (d) show satellite images in Austin (TX) on January 23, 2017 and July 5, 2017. Table 2-1 shows the 5 atmospheric conditions of 25 locations. The average aerosol optical depth (AOD) ranged from 0.08 to 0.31. The average ratio of pixels on satellite imagery that were covered by clouds to those which were uncovered (cloud) ranged from 0.07 to 0.35. Among all 25 locations, the average ground sampling distance of the image acquisition (gsd), the average water

vapor concentration used (water vapor), and the average ozone concentration used (ozone) were around 3.85 m, 1.5 g/cm^3, and 0.30 cm-atm, respectively.

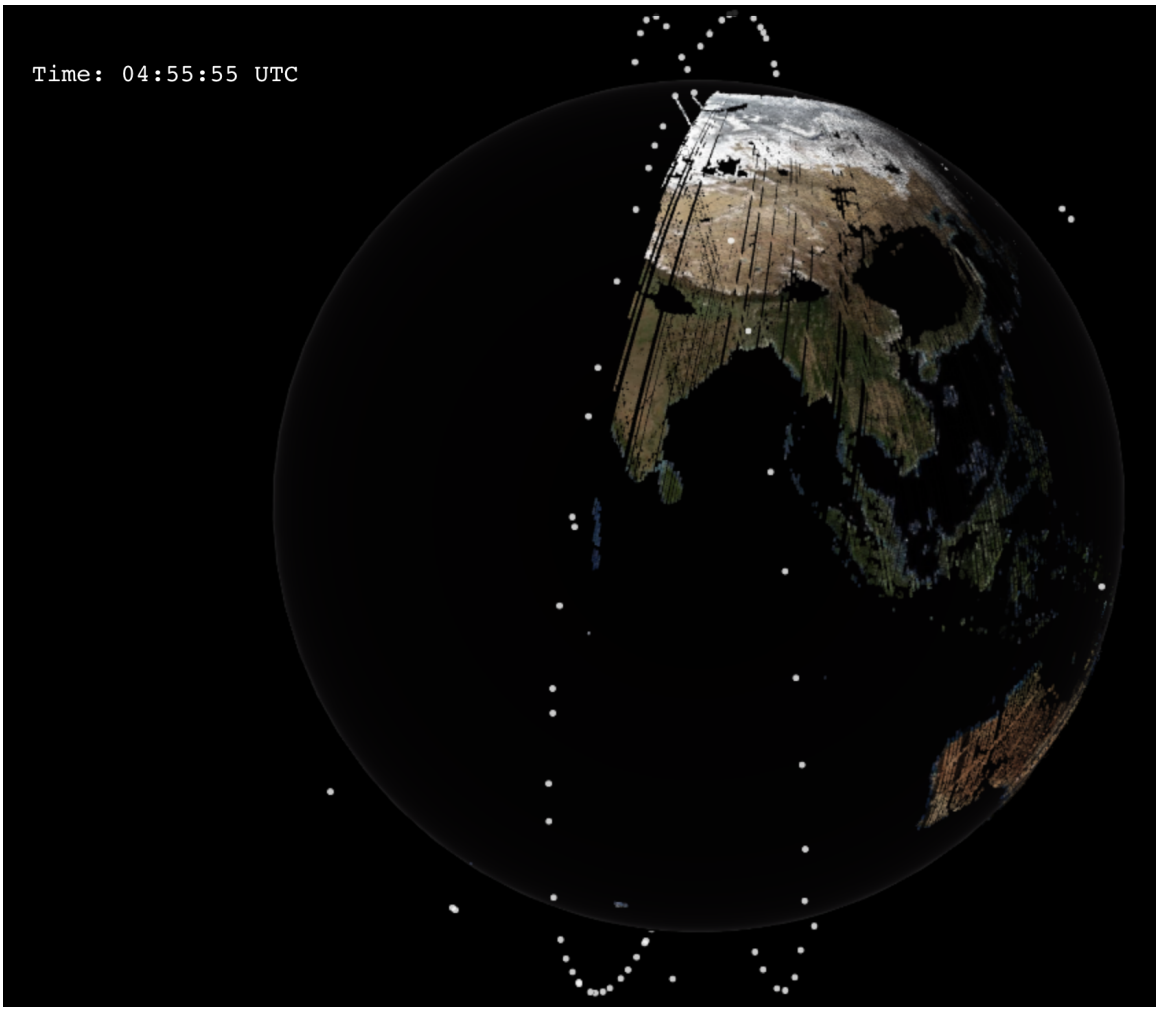
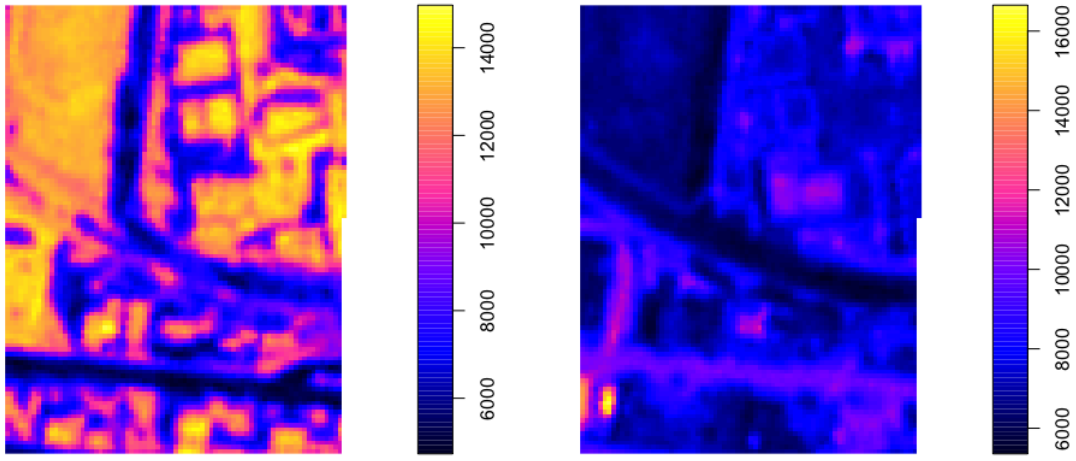


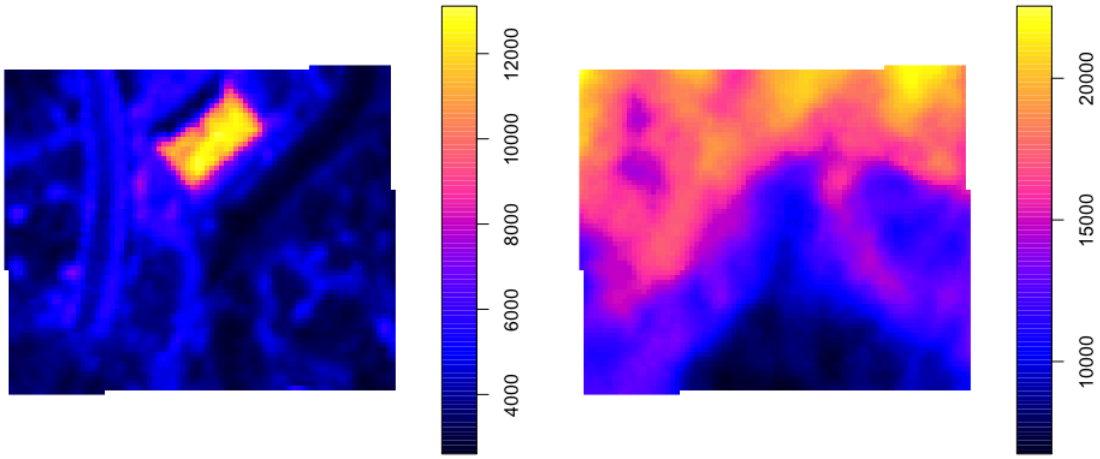
Figure 2-2: Illustration of PlanetScope satellites rotating around the Earth. The white circles represent the satellites, and the areas have covered by satellites are in color (photo source: https://www.planet.com/our-constellations/).

2.2 EPA Data

We acquired daily $PM_{2.5}$ measurements from EPA Air Quality System (https://www.epa.gov/) across the U.S. from January 2017 to October 2021. We did not include those collected before 2017 since the PlanetScope satellite product was at the early



(a) Satellite Image for Akron, 01/06/2017 **(b)** Satellite Image for Akron, 07/03/2017



(c) Satellite Image for Austin, 01/23/2017 **(d)** Satellite Image for Austin, 07/05/2017

Figure 2-3: Satellite Imaginary from Akron and Austin

stage of deployment and not working at full capacity. We matched each satellite image to its corresponding EPA monitor by longitude and latitude, and used EPA Federal Reference Method $PM_{2.5}$ as the validation data. Table 2-1 presents the summary statistics of these 25 locations. The average $PM_{2.5}$ concentrations ranged from 6.95 μg m^{-3} to 14.25 μg m^{-3} (standard deviation ranged from 3.33 μg m^{-3} to 13.36 μg m^{-3}). Figure 2-4 (a) shows the time series of $PM_{2.5}$ concentrations for Akron (OH) from

January 2017 to October 2021. There were some seasonal patterns, with relatively high values in summer months and winter months. There were some unexpected high $PM_{2.5}$ concentrations in 2020 summer, 2020 winter, and 2021 summer. Figure 2-4 (b) shows the time series of $PM_{2.5}$ concentrations for Austin (TX) from January 2017 to October 2021. There were also some seasonal patterns, with relatively high values in summer months. There were some unexpected high $PM_{2.5}$ concentrations in 2018 summer and 2020 summer.

Location	City/County	State	# Obs	mean($PM_{2.5}$) [$\mu g\ m^{-3}$]	std($PM_{2.5}$) [$\mu g\ m^{-3}$]	AOD	cloud	gsd [m]	water vapor [g/cm^3]	ozone [cm-atm]
Akron	city	Ohio	1396	9.40	5.14	0.12	0.26	3.85	1.34	0.32
Austin	city	Texas	1349	9.83	5.02	0.11	0.29	3.84	2.07	0.28
Bronx	county	New York	1247	8.38	4.90	0.08	0.22	3.85	1.41	0.32
Broward	county	Florida	1457	6.95	3.42	0.11	0.29	3.82	2.34	0.27
Cedar Rapids	city	Iowa	1534	8.93	4.87	0.12	0.16	3.87	1.41	0.32
Chicago	city	Illinois	1261	8.36	4.55	0.15	0.23	3.86	1.39	0.32
Essex	county	Maryland	166	8.03	6.09	0.09	0.23	3.84	1.49	0.32
Fort Collins	city	Colorado	1314	7.92	7.35	0.11	0.10	3.83	0.80	0.30
Los Angeles	city	California	1456	12.84	8.10	0.08	0.18	3.84	1.41	0.29
Merced	city	California	1382	13.67	13.36	0.16	0.15	3.84	1.48	0.29
Oakland	city	California	1615	10.35	12.34	0.12	0.24	3.84	1.33	0.30
Oldtown	county	Maryland	1425	8.57	4.90	0.10	0.22	3.87	1.63	0.32
Orlando	city	Florida	1223	6.96	3.33	0.31	0.27	3.82	2.14	0.28
Phoenix	city	Arizona	1456	8.17	6.06	0.10	0.11	3.82	1.51	0.29
Pittsburgh	city	Pennsylvania	1326	14.25	9.63	0.13	0.29	3.85	1.40	0.31
Queens	county	New York	1325	7.09	4.71	0.08	0.21	3.84	1.47	0.32
Raleigh	city	North Carolina	1347	9.01	3.93	0.13	0.22	3.84	1.81	0.30
Salt Lake	city	Utah	1662	7.72	6.17	0.12	0.15	3.84	0.81	0.30
Seattle	city	Washington	1337	6.54	9.26	0.09	0.35	3.87	1.10	0.32
St. Louis	city	Missouri	1515	9.62	4.57	0.11	0.18	3.86	1.65	0.31
St. Paul	city	Minnesota	1309	7.94	5.04	0.09	0.15	3.86	1.15	0.32
Terre Haute	city	Indiana	1381	9.41	5.27	0.14	0.17	3.84	1.57	0.31
Timonium	county	Maryland	1002	8.49	4.33	0.12	0.26	3.85	1.57	0.31
Tranquillity	county	California	1580	8.78	11.81	0.14	0.08	3.84	1.46	0.30
Yuma	city	Arizona	1488	8.60	3.73	0.13	0.07	3.82	1.71	0.29

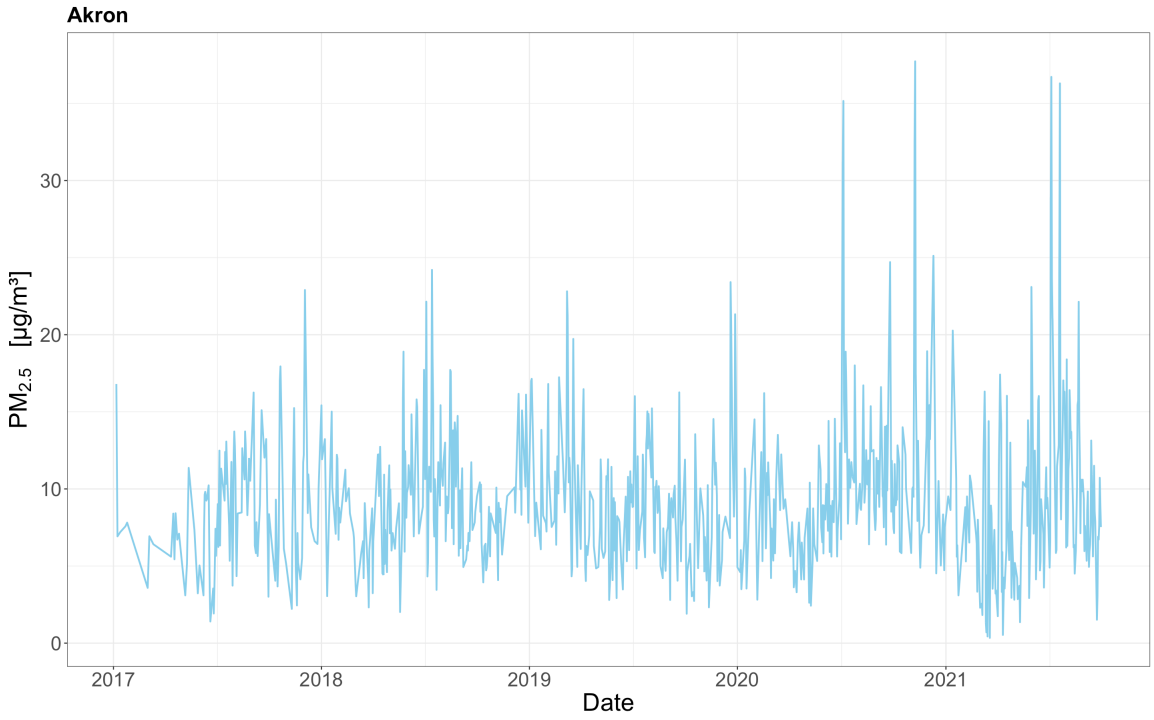
Table 2-1: Basic and Atmospheric Information of 25 Locations. Number of observations represents the number of data that have all satellite imagery, EPA measurement, and meteorological information available. mean($PM_{2.5}$) is the average $PM_{2.5}$ concentration collected from EPA monitors during January 2017 to October 2021. std($PM_{2.5}$) is the standard deviation of $PM_{2.5}$ concentration collected from EPA monitors during January 2017 to October 2021. AOD represents the average aerosol optical depth. Cloud represents the average ratio of pixels on satellite imagery that are covered by clouds to those which are uncovered. Gsd represents the average ground sampling distance of the image acquisition. Water vapor represents the average water vapor concentration used. Ozone represents the average ozone concentration used. *Note that Essex has significantly fewer observations because it does not have daily observations.*

2.3 Meteorological Measurements

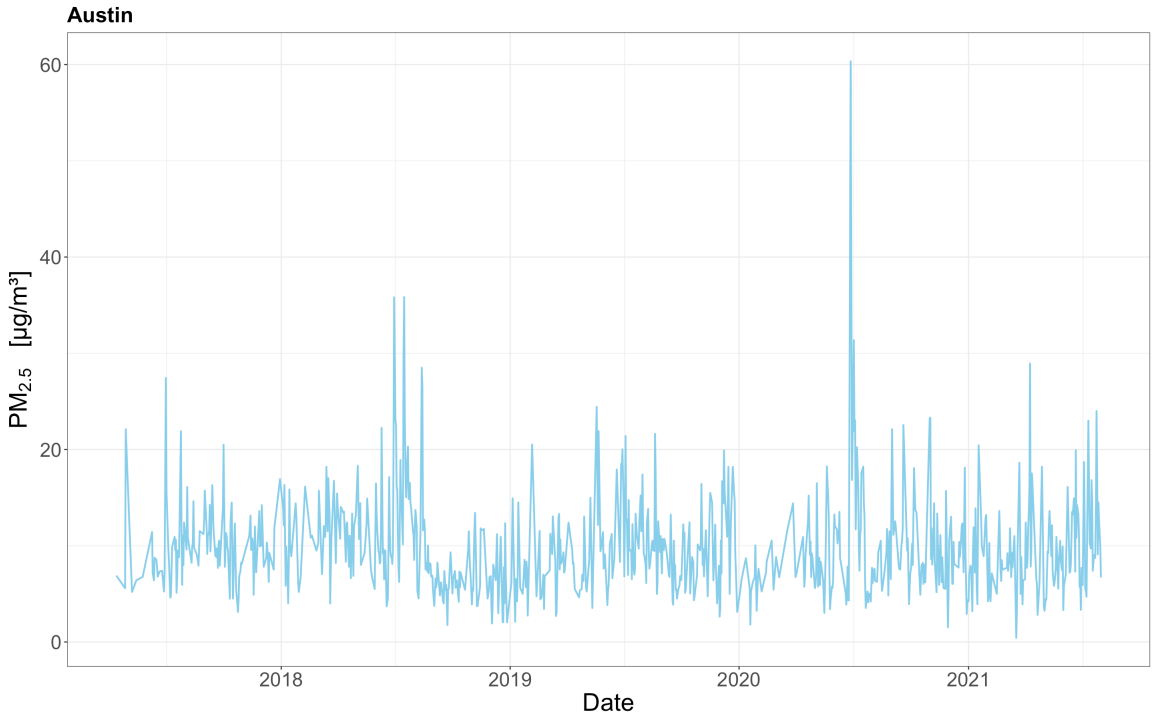
Given the potential differences between U.S. regions, it is important to characterize the factors that contribute to $PM_{2.5}$ concentrations. Meteorological data have been used as predictors in most of $PM_{2.5}$ prediction models, and are considered to have unique advantages in retrieving historical features of $PM_{2.5}$ (Akyüz and Çabuk, 2009). The meteorological data were downloaded from NASA POWER portal (https://power.larc.nasa.gov/data-access-viewer/). We matched meteorological data to image-$PM_{2.5}$ pairs by corresponding location and date. Table 2-2 shows the 6 meteorological measurements of 25 locations. The average air temperature at 2 meters (temp) ranged from 8.02 ° C to 25.92 ° C. The average dew point temperature at 2 meters (dew point temp) ranged from -1.02 ° C to 21.02 ° C. The average relative humidity (rh) ranged from 31.30 % to 79.83 %. The average daily precipitation (precip) ranged from 0.11 mm/day to 3.77 mm/day. The average surface pressure (pressure) ranged from 82.19 kPa to 101.66 kPa. The average wind speed (ws) at 10 meters ranged from 1.78 m/s to 4.97 m/s.

Location	temp [° C]	dew point temp [° C]	rh [%]	precip [mm/day]	pressure [kPa]	ws [m/s]
Akron	13.42	8.45	75.19	1.40	97.85	2.61
Austin	21.88	14.09	65.96	1.19	99.46	4.01
Bronx	12.33	8.17	78.32	1.45	100.87	3.00
Broward	25.19	20.21	75.02	2.30	101.66	4.26
Cedar Rapids	25.92	21.02	75.34	3.77	101.62	4.16
Chicago	13.20	9.46	79.14	1.65	99.57	4.96
Essex	13.31	8.92	76.99	1.34	100.83	3.77
Fort Collins	11.90	-1.02	46.26	0.72	82.19	3.72
Los Angeles	18.45	6.90	54.44	0.40	96.94	2.72
Merced	18.63	7.53	55.90	0.52	96.92	2.67
Oakland	14.87	9.14	72.74	0.44	100.56	3.55
Oldtown	12.96	8.17	76.00	1.62	96.64	2.02
Orlando	23.13	18.07	76.04	2.79	101.53	3.55
Phoenix	24.06	3.14	31.30	0.51	95.55	2.94
Pittsburgh	13.55	8.74	76.10	1.34	97.94	1.94
Queens	13.75	10.18	79.83	1.72	101.66	4.97
Raleigh	16.54	11.42	75.27	1.66	100.38	2.05
Salt Lake	11.12	-0.61	51.26	0.67	82.66	2.41
Seattle	11.95	8.13	79.63	1.33	100.51	1.78
St. Louis	15.38	10.77	76.25	1.72	99.95	3.44
St. Paul	8.02	3.72	77.21	1.07	98.44	3.80
Terre Haute	13.83	9.39	76.79	1.50	99.64	3.97
Timonium	14.21	9.60	76.34	1.59	100.02	3.75
Tranquillity	20.47	4.72	44.44	0.25	99.82	3.23
Yuma	24.52	5.43	35.87	0.11	99.54	3.37

Table 2-2: Meteorological Conditions of 25 Locations. Temp is the average air temperature at 2 meters. Dew point temp is the average dew point temperature at 2 meters. Rh is the average relative humidity. Precip is the average daily precipitation. Pressure is the average surface pressure. Ws is the average wind speed at 10 meters.



(a) Time Series of $PM_{2.5}$ Concentrations for Akron, January 2017 - October 2021



(b) Time Series of $PM_{2.5}$ Concentrations for Austin, January 2017 - October 2021

Figure 2-4: Time Series of $PM_{2.5}$ Concentrations for Akron and Austin, January 2017 - October 2021

Chapter 3

Methods

All analyses were conducted in R. Satellite imagery of 25 locations were downloaded from Planet (https://www.planet.com/) as GeoTIFF files. We used `readGDAL` function in `rgdal` package to read satellite imagery files, `xgboost` package for XGBoost approach, and `keras` package for CNN approach.

3.1 Extreme Gradient Boosting for All Locations

XGboost is a gradient descent algorithm that minimizes the loss function by tuning parameters iteratively, where a loss function is used to measure how far an estimated value is from its true value. Common loss functions are Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). XGBoost is a supervised learning algorithm that implements a boosting process to produce accurate predictions. Supervised learning refers to the task of inferring a predictive model from a set of labeled training examples. This predictive model can then be applied to new unseen examples or the testing examples. The data format of XGBoost input is usually a matrix with each row representing an observation and each column representing a feature.

In this analysis, we used `vtreat` package to convert the data frame into the required matrix format. We also incorporated a 5-fold cross validations with 1,000 trees. The hyperparameters we tuned when implementing XGBoost were:

(1) maximum depth of a tree (max_depth): controls the complexity of the boosted ensemble. Default value is 6, and we set it to 6, 8, and 10.

(2) learning rate (eta): controls how quickly the algorithm proceeds down the gradient descent. If the learning rate is too small, the algorithm may take several iterations to find the minimum. If the learning rate is too large, it might jump across the minimum and end up further away than the starting point. Default value is 0.3, and we set it to 0.1, 0.2, and 0.3.

(3) minimum sum of instance weight needed in a child (min_child_weight): controls the minimum number of instances needed to be in each node, where a greater value leads to a more conservative algorithm. Default value is 1, and we set it to 2, 4, and 6.

(4) subsample ratio of the training instance (subsample): controls what proportion of the available training observations are used, where using less than 100% of observations means implementing stochastic gradient descent. This parameter is used to minimize overfitting and to avoid getting stuck in a local minimum or plateau of the loss function gradient. Default value is 1, and we set it to 0.6, 0.8, and 1.

(5) subsample ratio of columns (colsample_bytree): controls proportion of columns are used when constructing each tree. Default value is 1, and we set it to 0.6, 0.8, and 1.

To find the optimal parameters, we created a hyperparameters grid consisting of 243 different hypoerparameter combinations (*i.e.* 3 possible values for each parameter $\rightarrow$ run $3^5 = 243$ models). The optimal tree booster we built had learning rate of 0.1, and maximum depth of each tree of 8. The minimum number of instances in each node was 6, with 80% of the training instance and 80% of columns of each tree.

For XGBoost model, after reading in each image, we split it into 16 sections (shown in Figure 3-1) and extracted the mean and standard deviation of each band within each section. Thus, we obtained 128 summary statistics for each image (*i.e.* 16 sections $\times$

4 bands $\times$ 2 summary statistics = 128).

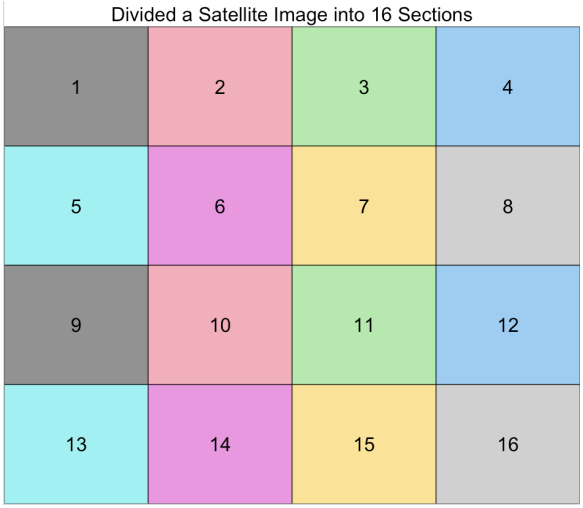


Figure 3-1: Illustration of Splitting an Image

Atmospheric information (AOD, cloud, gsd, water vapor, and ozone) and meteorological conditions (temp, dew point temp, rh, pressure, precip and ws), which are identified as useful parameters in predicting $PM_{2.5}$, were also included as covariates.

We combined all observations from 25 locations. To better capture the seasonality patterns, we included year and month as two categorical variables. So for each observation, there were 128 summary statistics from satellite imagery, 5 atmospheric conditions, 6 meteorological information, 2 categorical variables, and 1 EPA Federal Reference Method $PM_{2.5}$ concentrations. We also removed any observations that had missing bands information in certain sections of the satellite images (*i.e.* containing one or more `NA`'s in 128 summary statistics from satellite imagery). We further standardized the 128 summary statistics. Thus, we obtained a 31,565 $\times$ 142 data frame (31,565 observations $\times$ (141 covariates + 1 response)). These 31,565 observations were randomly divided into 80% and 20% splits for a training set (25,252 $\times$ 142) and a testing set (6,313 $\times$ 142).

3.2 Deep Convolution Neural Network for Each Location

CNN is one of the Deep Neural Networks that can recognize and extract particular features from images and are widely used for analyzing visual images. The two main parts of a CNN architecture are

(1) a convolution tool: used for feature extraction, where it identifies and separates the various features of the image;

(2) a connected layer: used for prediction, where it uses output from the convolution process and makes predictions based on the extracted features.

In this analysis, we built a sequential model using a linear stack of layers. Our architecture included two convolution 2d layers and max pooling pairs followed by a flatten layer, which is usually used as a connection between convolution and the dense layers.

The first layer consisted of an input image with dimensions of $i \times j \times 4$, where $i \times j$ is the size of the image, and 4 represents 4 bands. It was convolved with 32 filters of size 3×3. The second layer was a max pooling operation, a variant of subsampling, where the maximum pixel value falling within the receptive field of a unit within a subsampling layer is taken as the output. This layer had a size of 2 $\times$ 2 and slid across the input image, outputting an average of the pixels within the receptive field of the kernel. Similarly to the first layer, the third layer also involved in a convolution operation with 64 filters of size 3×3, followed by a fourth pooling layer with same size of 2×2. One of the most important parameters of the CNN model is the activation function, which is used to learn the relationship between variables within the architecture. Here we used ReLU, the rectified linear unit activation function. The fifth layer was a flatten layer that took an input shape and flattened the input image data into a one-dimensional array. The last dense layer was an output layer

containing a single neuron with a linear activation function.

```
model <- keras_model_sequential() %>%
  layer_conv_2d(filters = 32, kernel_size = c(3, 3),
              input_shape = c(i,j,4),
               activation ="relu") %>%
  layer_max_pooling_2d(pool_size = c(2,2)) %>%
  layer_conv_2d(filters = 64, kernel_size = c(3,3),
  activation ="relu") %>%
  layer_max_pooling_2d(pool_size = c(2,2)) %>%
  layer_flatten() %>%
  layer_dense(units = 256, activation ="relu") %>%
  layer_dense(units = 1)
```

The batch size and the number of epochs are important hyperparameters in CNN. The batch size is a parameter of gradient descent that controls the number of training samples to loop through before updating the model's internal parameters. At the end of the batch, predictions are compared to the expected output variables and an error is calculated. Based on the error, an updated algorithm is used to improve the model performance. If batch size equals to the size of training set, the learning algorithm is called batch gradient descent. If batch size is one sample, the learning algorithm is called stochastic gradient descent. Similarly, if the batch size is greater than 1 but less than the size of the training set, the learning algorithm is called mini-batch gradient descent. Common batch sizes are 32, 64, and 128. In this analysis, we used mini-batch gradient descent learning algorithm with batch size of 32.

The number of epochs is a parameter of gradient descent that controls the number of complete processes going through the training set. One epoch means that each sample in the training set has one chance to update the model's internal parameters. Learning curves are commonly used to diagnose whether the model has over learned, under learned, or is suitably fit to the training set, where the x-axis represents the number of epochs and the y-axis represents the mean squared error. In this analysis, the number of epochs that we used was 50, since the mean squared error curve of the training set decreased to a point of stability around 50 epochs.

For CNN model, we used the information of four bands directly without standard-

ization. One limitation of `keras` package is that the first layer requires a specified image size. Since the sizes of satellite imagery vary in 25 locations, we decided to fit a CNN model for each location. Different from XGBoost model, removing any observations that had missing bands information will change the sizes of satellite imagery and thus violate the pre-set dimensions of images. Thus, we replaced all `NA`'s with 0's.

Another limitation of `keras` package is that it cannot work with an image and additional covariates at the same time. So we only used extracted features from satellite imagery and did not include any atmospheric conditions or meteorological information.

As a result, we formulated n_k arrays with dimension of $i_k \times j_k \times 4$, where n_k is the number of observations in location k, $i_k \times j_k$ is size of the image in location k, and 4 represents 4 is the number of bands (RGB, near-infrared). Within each location, n_k arrays were randomly divided into $\sim 80\%$ and $\sim 20\%$ splits for training set and testing set, along with the corresponding EPA Federal Reference Method $PM_{2.5}$ concentrations as validation data.

3.3 Comparison and Combination

3.3.1 Comparison: two models for Each Location

Besides combining all the observations and fitting one XGBoost model, we also would like to see how each XGBoost model performed within each location.

Due to the limitations of `keras` package, we modified our XGBoost to achieve a fair comparison. First, we did not standardize the extracted 128 summary statistics from satellite imagery. Second, for those observations that had missing bands information in certain sections of the satellite images, we replaced `NA`'s with 0's. Finally, we did not include any atmospheric conditions or meteorological information. Thus, we obtained

an $n_k \times$ 129 data frame in each location, where n_k is the number of observations in location k, 129 is consisted of 128 covariates plus 1 response. Within each location, observations were randomly divided into $\sim$ 80% and $\sim$ 20% splits for a training set and a testing set. We compared two model performances within each location.

3.3.2 Combination: CNN-XGBoost Pipeline for All Locations

Since the CNN model only used extracted features from satellite imagery and did not include any atmospheric conditions or meteorological information, which are considered to be important factors in affecting model performance (Stowell et al., 2020), we decided to implement a CNN-XGBoost pipeline.

Within each city, the observations were randomly divided into $\sim$ 80% and $\sim$ 20% splits for a training set and a testing set. During the first stage of our pipeline, we employed CNN for each location and got predictions from both the training set and the testing set. We combined the predictions from 25 training sets and 25 testing sets separately. In the second stage, we used the CNN predictions, 5 atmospheric conditions, 6 meteorological information, and 2 categorical variables as the inputs of XGBoost model to make final predictions. Thus, we obtained a training set of a 26,852 $\times$ 15 data frame (26,852 observations $\times$ (14 covariates + 1 response)), and a testing set of a 6,710 $\times$ 15 data frame (6,710 observations $\times$ (14 covariates + 1 response)) as XGBoost inputs.

3.4 Evaluation Criteria

To measure the accuracy of our models, we used Root Mean Square Error (RMSE) and coefficient of determinant (R^2).

RMSE is a standard way to measure the error of a model in predicting continuous

data. It is non-negative, and a lower value suggests a better fit. It is calculated as

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{3.1}$$

where n is the number of observations, $\hat{y}_i$ is the predicted $PM_{2.5}$ value for observation i, and y_i is the truth for observation i.

R^2 is the percentage of the variation of actual values from the mean value that can be explained by the regression model. It is the ratio of the sum of squares regression (SSR) and the sum of squares total (SST). R^2 is used to measure the goodness of fit and a higher value suggests a better fit. In most cases it lies between 0 and 1 inclusively, but it could be negative whenever the model's predictions are worse than a constant function that always predicts the mean of the data. It is calculated as

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}{\sum_{i=1}^{n} \frac{(y_i - \mu_y)^2}{n}} = 1 - \frac{(RMSE)^2}{Var(y)} \tag{3.2}$$

Chapter 4

Results

4.1 XGBoost

There were 31,565 available observations, where 18 observations with $PM_{2.5}$ concentrations greater than 100 $\mu g\ m^{-3}$. The 31,565 observations had an average $PM_{2.5}$ of 9.08 $\mu g\ m^{-3}$ (standard deviation (SD) = 7.30 $\mu g\ m^{-3}$). The training set had an average $PM_{2.5}$ of 9.12 $\mu g\ m^{-3}$ (SD = 7.45 $\mu g\ m^{-3}$). The testing set had a slightly lower average $PM_{2.5}$ of 8.94 $\mu g\ m^{-3}$ (SD = 6.68 $\mu g\ m^{-3}$). The RMSE of testing set was 3.98 $\mu g\ m^{-3}$ and R^2 was 0.65.

Table 4-1 shows the RMSE of each month for the testing set. August had the highest average $PM_{2.5}$ concentrations of 11.12 $\mu g\ m^{-3}$ (SD = 9.68 $\mu g\ m^{-3}$). It also had the second highest R^2 of 0.76 and the highest RMSE of 4.73 $\mu g\ m^{-3}$, following by December and November. Although it is not common to see high RMSE with high R^2, August had a much higher standard deviation for $PM_{2.5}$ concentrations than other months, indicating the observations were widely spread out from the average $PM_{2.5}$ concentrations and thus leading to higher prediction errors. Also, since R^2 is influenced by variance, as long as the RMSE is much smaller than the standard deviation of $PM_{2.5}$ concentrations, we would have a high R^2. June had the lowest RMSE of 2.95 $\mu g\ m^{-3}$, with R^2 of 0.43. Table 4-2 shows the RMSE of each year for the testing set. 2020 had the highest average $PM_{2.5}$ concentrations of 9.51 $\mu g\ m^{-3}$

(SD = 9.50 μg m^{-3}). It had the highest R^2 of 0.75 and the highest RMSE of 4.75 μg m^{-3}, following by 2017 and 2021. Similarly, although it seems contradictory to have high RMSE with high R^2, 2020 had a much higher standard deviation than other years. 2019 had the lowest average $PM_{2.5}$ concentrations of 7.92 μg m^{-3} (SD = 4.72 μg m^{-3}). It had the lowest RMSE of 3.30 μg m^{-3} and the lowest R^2 of 0.51. From these two tables, we concluded that when both the mean and the standard deviation of true $PM_{2.5}$ concentrations were high, XGBoost model tended to have high RMSE but high R^2 as well.

Figure 4-1 is a scatter plot of the XGBoost predicted $PM_{2.5}$ against EPA $PM_{2.5}$, which shows a linear trend between predicted $PM_{2.5}$ and EPA $PM_{2.5}$. The fitted regression line (blue dashed line) overlapped with the 45° diagonal line (red solid line), suggesting that the XGBoost predictions were relatively consistent with the actual EPA measurements.

Month	# Obs	mean($PM_{2.5}$) [μg m^{-3}]	std($PM_{2.5}$) [μg m^{-3}]	RMSE [μg m^{-3}]	R^2
August	651	11.12	9.68	4.73	0.76
December	449	10.45	6.65	4.50	0.54
November	448	9.47	6.49	4.35	0.55
October	482	8.84	7.32	4.33	0.65
February	360	9.04	6.78	4.24	0.61
September	618	9.62	9.26	4.23	0.79
July	699	9.90	5.93	4.10	0.52
January	397	8.53	4.99	3.87	0.40
March	492	7.05	5.03	3.83	0.42
April	539	7.43	4.39	3.24	0.45
May	537	7.51	4.26	3.09	0.47
June	640	7.81	3.92	2.95	0.43

Table 4-1: XGBoost RMSE by month of testing set. Number of observations represents the number of observations in each month of testing set. mean($PM_{2.5}$) is the average $PM_{2.5}$ of each month collected from EPA monitors during January 2017 to October 2021. std($PM_{2.5}$) is the standard deviation of $PM_{2.5}$ of each month collected from EPA monitors during January 2017 to October 2021. RMSE is the root mean square error of each month. R^2 is the R^2 of each month.

Year	# Obs	mean($PM_{2.5}$) [$\mu g\ m^{-3}$]	std($PM_{2.5}$) [$\mu g\ m^{-3}$]	RMSE [$\mu g\ m^{-3}$]	R^2
2017	818	8.87	5.84	4.03	0.52
2018	1427	9.22	5.70	3.78	0.56
2019	1493	7.92	4.72	3.30	0.51
2020	1450	9.51	9.50	4.75	0.75
2021	1124	9.28	6.05	3.93	0.58

Table 4-2: XGBoost RMSE by year of testing set. Number of observations represents the number of observations in each year of testing set. mean($PM_{2.5}$) is the average $PM_{2.5}$ of each year collected from EPA monitors during January 2017 to October 2021. std($PM_{2.5}$) is the standard deviation of $PM_{2.5}$ of each year collected from EPA monitors during January 2017 to October 2021. RMSE is the root mean square error of each year. R^2 is the R^2 of each year.

4.2 Cloud Cover Issue

One of the major issues of the satellite-based methods is that they are dependent on availability of clear sky, and can provide clear images only when there are no clouds. Since $PM_{2.5}$ concentrations cannot be estimated from satellite observations under cloudy conditions or bright surfaces such as snow or ice, we gathered all satellite imagery that had cloud coverage below 5%. To do this, we used one of the atmospheric information metrics, cloud, obtained from Planet. The 16,428 observations had an average $PM_{2.5}$ of 9.02 $\mu g\ m^{-3}$ (SD = 6.00 $\mu g\ m^{-3}$). The training set had the same average $PM_{2.5}$ as the overall data, and slightly higher standard deviation of 6.04$\mu g\ m^{-3}$. The testing set had a slightly lower average $PM_{2.5}$ of 9.00 $\mu g\ m^{-3}$ (SD = 5.84 $\mu g\ m^{-3}$). The RMSE of testing set was 3.67 $\mu g\ m^{-3}$ and R^2 was 0.60. Table 4-3 shows that eliminating all satellite imagery with cloud coverage above 5% was not the best approach since the resulting data set was not a good representation of the overall data. For example, 80% of the observations from Yuma (CA) were selected, while only 19% of the observations from Broward (FL) were selected. Table 4-4 provides additional evidence. For example, 57% observations from June, July, September, and October were selected, while only 43% observations from January were selected.

Due to the complexity of terrains across the U.S., the atmospheric stability during

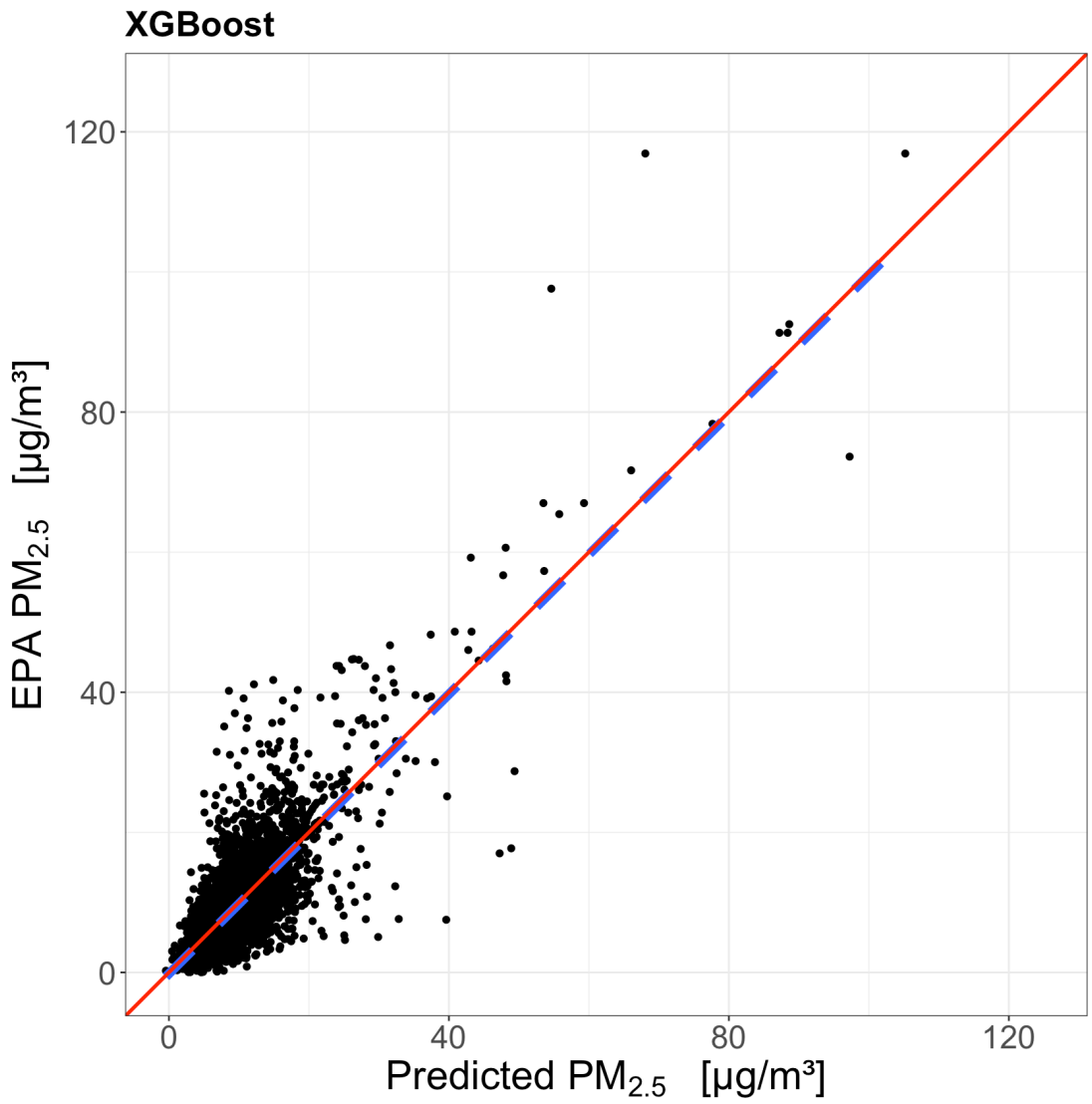


Figure 4-1: Scatter Plot of XGboost Predicted $PM_{2.5}$ against EPA $PM_{2.5}$. The black points are the data points in the testing set. The red solid line represents the 45° diagonal line, while the blue dashed line represents the fitted regression line.

winter time affects local $PM_{2.5}$ concentrations variously (Karandana Gamalathge and Green, 2017). Table 4-1 shows that winter months (December, November, and January) had relatively high average $PM_{2.5}$ values and variability. If the majority observations of winter months were removed due to high cloud coverage, the overall predictions would likely to be underestimated. Table 4-5 suggests that Broward (FL) had relatively low average $PM_{2.5}$ values and variability. If the majority observations of Broward were removed, then the overall predictions would likely to be overestimated.

To avoid the imbalance of data, we decided to use all observations for the following analysis.

Location	Total # Obs	# Obs with cloud < 0.05 (% of total obs)
Akron	1322	493 (37%)
Austin	1284	467 (37%)
Bronx	1247	595 (51%)
Broward	1372	265 (19%)
Cedar Rapids	1448	848 (59%)
Chicago	1188	531 (45%)
Essex	166	75 (45%)
Fort Collins	1230	822 (67%)
Los Angeles	1366	834 (61%)
Merced	1297	831 (64%)
Oakland	1505	768 (51%)
Oldtown	1425	587 (45%)
Orlando	1141	370 (32%)
Phoenix	1385	989 (71%)
Pittsburgh	1260	497 (39%)
Queens	1264	658 (52%)
Raleigh	1266	690 (55%)
Salt Lake	1545	834 (54%)
Seattle	1268	439 (35%)
St Louis	1433	781 (55%)
St Paul	1209	754 (62%)
Terre Haute	1316	655 (50%)
Timonium	948	416 (43%)
Tranqullity	1481	1087 (73%)
Yuma	1410	1133 (80%)

Table 4-3: XGBoost Accounting for Cloud Cover Issue by Location. Number of observations represents the total number of observations in each location, collected from EPA monitors during January 2017 to October 2021. Number of observations with cloud < 0.05 represents the number of observations that have cloud coverage below 5% in each location, and its corresponding percentage of total number of observations. *Note that we removed any observations that had missing bands information in certain sections of the satellite images, so there were fewer observations in each location than that of in Chapter 2.*

Month	# Obs	# Obs with cloud < 0.05 (% of total obs)
January	397	171 (43%)
February	360	164 (46%)
March	492	254 (52%)
April	539	288 (53%)
May	537	291 (54%)
June	640	366 (57%)
July	699	395 (57%)
August	651	310 (48%)
September	618	355 (57%)
October	482	273 (57%)
November	448	218 (49%)
December	449	200 (45%)

Table 4-4: XGBoost Accounting for Cloud Cover Issue by Month. Number of observations represents the total number of observations in each month, collected from EPA monitors during January 2017 to October 2021. Number of observations with cloud < 0.05 represents the number of observations that have cloud coverage below 5% in each month, and its corresponding percentage of total number of observations.

4.3 Comparison of Two Models

After replacing `NA`'s with 0's for those observations with missing bands information in certain sections of the satellite images, there were 33,562 observations. Table 4-5 shows the comparison of two models for each location. The RMSE of XGBoost was lower than that of CNN in every location, suggesting that XGBoost provided a more robust estimation. Merced (CA) and Tranquallity (CA) had relatively higher R^2, but they also had relatively higher RMSE. Some locations had quite high CNN RMSE, so we did not include the CNN R^2 in the table (*i.e.* those R^2 can be negative and thus are not informative). Figure 4-2 shows 25 time series plots of CNN predicted $PM_{2.5}$ against EPA $PM_{2.5}$. There were couple of negative predictions in almost every location, and some predictions were even below -10 $\mu g\ m^{-3}$ in Austin (TX), Cedar Rapids (IA), Chicago (IL), and Essex (MD). In addition, the predictions of Fort Collins (CO),

Oakland (CA), and Tranquillity (CA) were around 0. These unexpected behaviors suggested that the CNN model was not accurate and not able to capture the daily dynamic changes. Figure 4-3 shows 25 time series plots of XGBoost predicted $PM_{2.5}$ against EPA $PM_{2.5}$. In general, it performed better than CNN in every location and there was only one prediction below 0 (in Seattle, WA). It was able to capture most of the relatively high values, except the ones greater than 70 $\mu g\ m^{-3}$ in Los Angeles (CA), Phoenix (AZ), and Tranquillity (CA). Figure 4-4 shows 25 scatter plots of CNN predicted $PM_{2.5}$ against EPA $PM_{2.5}$, which indicates that most of the predictions overestimated the actual values. The majority of predictions were the same in Fort Collins (CO) and Tranquillity (CA), suggesting a different CNN architecture need to be used for depicting the changes. Figure 4-5 shows 25 scatter plots of XGBoost predicted $PM_{2.5}$ against EPA $PM_{2.5}$. While most of the predictions were consistent with the actual values, it was hard for XGboost to predict the relatively high values, such as the ones in Essex (MD), Los Angeles (CA), Oakland (CA), and Phoenix (AZ). All these tables and figures agree that XGBoost achieved better performances.

4.4 CNN-XGBoost Pipeline

The training set had an average $PM_{2.5}$ of 9.08 $\mu g\ m^{-3}$ (SD = 7.27 $\mu g\ m^{-3}$). The testing set had a slightly higher average $PM_{2.5}$ of 9.13 $\mu g\ m^{-3}$ (SD = 7.42 $\mu g\ m^{-3}$). The RMSE of testing set was 5.87 $\mu g\ m^{-3}$ and R^2 was 0.37. Figure 4-6 is a scatter plot of CNN-XGBoost predicted $PM_{2.5}$ against EPA $PM_{2.5}$, which shows a linear trend between predicted $PM_{2.5}$ and EPA $PM_{2.5}$. The fitted regression line (blue dashed line) was less steeper than the 45° diagonal line (red solid line), suggesting that the CNN-XGBoost predictions overestimated the actual EPA measurements.

Location	# obs	mean ($PM_{2.5}$) [$\mu g\ m^{-3}$]	std($PM_{2.5}$) [$\mu g\ m^{-3}$]	CNN RMSE [$\mu g\ m^{-3}$]	XGBoost RMSE [$\mu g\ m^{-3}$]	XGBoost R^2
Akron	279	9.53	4.92	6.17	3.93	0.36
Austin	270	10.16	4.88	8.09	4.36	0.20
Bronx	249	8.22	4.39	6.58	3.88	0.22
Broward	291	7.04	4.16	10.68	3.59	0.26
Cedar Rapids	307	8.93	4.99	11.01	4.32	0.25
Chicago	252	8.50	4.69	6.81	4.14	0.22
Essex	33	8.65	7.44	11.69	6.62	0.21
Fort Collins	263	7.48	6.28	9.76	5.18	0.32
Los Angeles	291	13.16	11.43	12.37	10.99	0.08
Merced	276	13.94	14.14	10.25	9.52	0.55
Oakland	323	9.64	8.92	13.15	6.33	0.50
Oldtown	285	8.47	5.04	5.39	4.38	0.24
Orlando	245	6.92	3.18	5.63	3.11	0.04
Phoenix	293	8.54	8.52	7.97	7.94	0.13
Pittsburgh	265	14.22	9.18	9.78	7.70	0.30
Queens	265	7.14	4.48	5.10	4.27	0.09
Raleigh	269	8.82	3.73	4.84	3.68	0.03
Salt Lake	332	7.77	5.90	6.07	5.44	0.15
Seattle	267	6.29	6.30	6.31	6.12	0.06
St Louis	303	9.63	4.50	4.94	4.28	0.10
St Paul	262	7.50	4.54	8.37	4.18	0.15
Terre Haute	267	6.29	6.30	6.31	6.12	0.06
Timonium	200	8.89	4.82	6.68	4.38	0.17
Tranqullity	316	9.45	13.98	13.58	9.19	0.57
Yuma	298	8.86	3.98	5.26	3.91	0.03

Table 4-5: Model Comparison for Each Location. Number of observations represents the number of observations in testing set. mean($PM_{2.5}$) is the average $PM_{2.5}$ concentrations in the testing set collected from EPA monitors during January 2017 to October 2021. std($PM_{2.5}$) is the standard deviation of $PM_{2.5}$ concentrations in the testing set collected from EPA monitors during January 2017 to October 2021. CNN RMSE is the RMSE of the testing set from CNN model in each location. XGBoost RMSE is the RMSE of the testing set from XGBoost model in each location. XGBoost R^2 is the R^2 of the testing set from XGBoost model in each location. *Note that the CNN R^2's were not included in the table since those R^2's were negative and thus are not informative.*

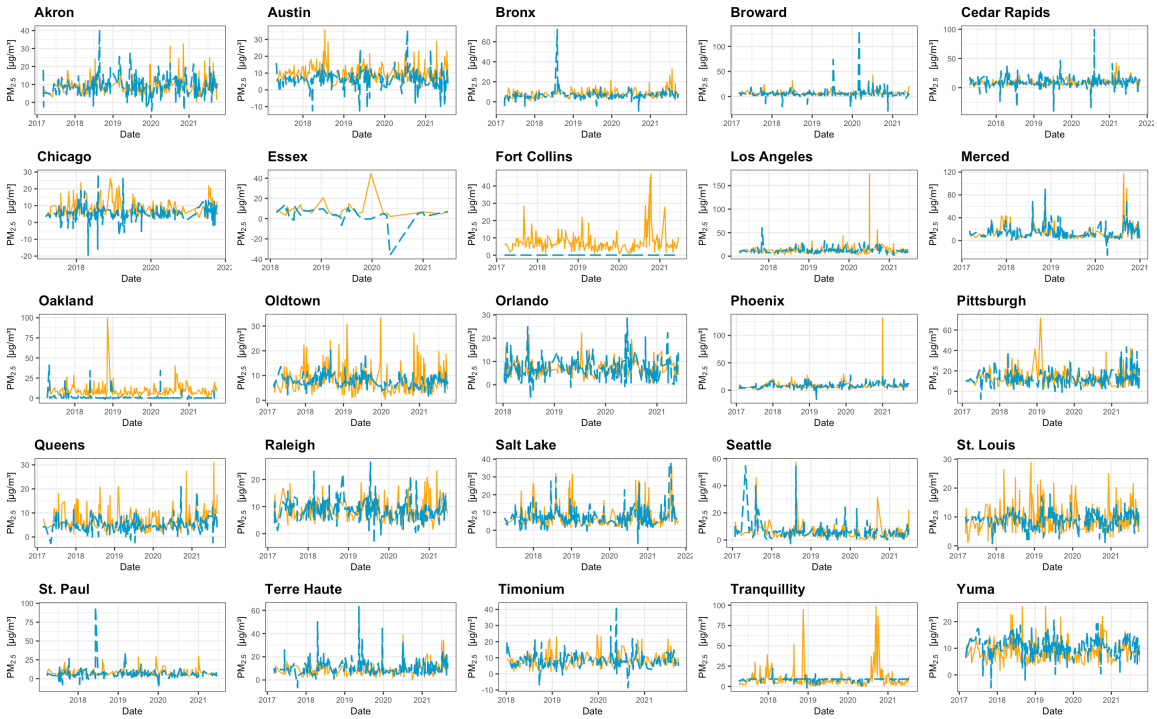


Figure 4-2: 25 Time Series Plots of CNN Predicted $PM_{2.5}$ against EPA $PM_{2.5}$. The orange solid line represents the actual EPA measurements, while the blue dashed line represents the predictions from CNN model.

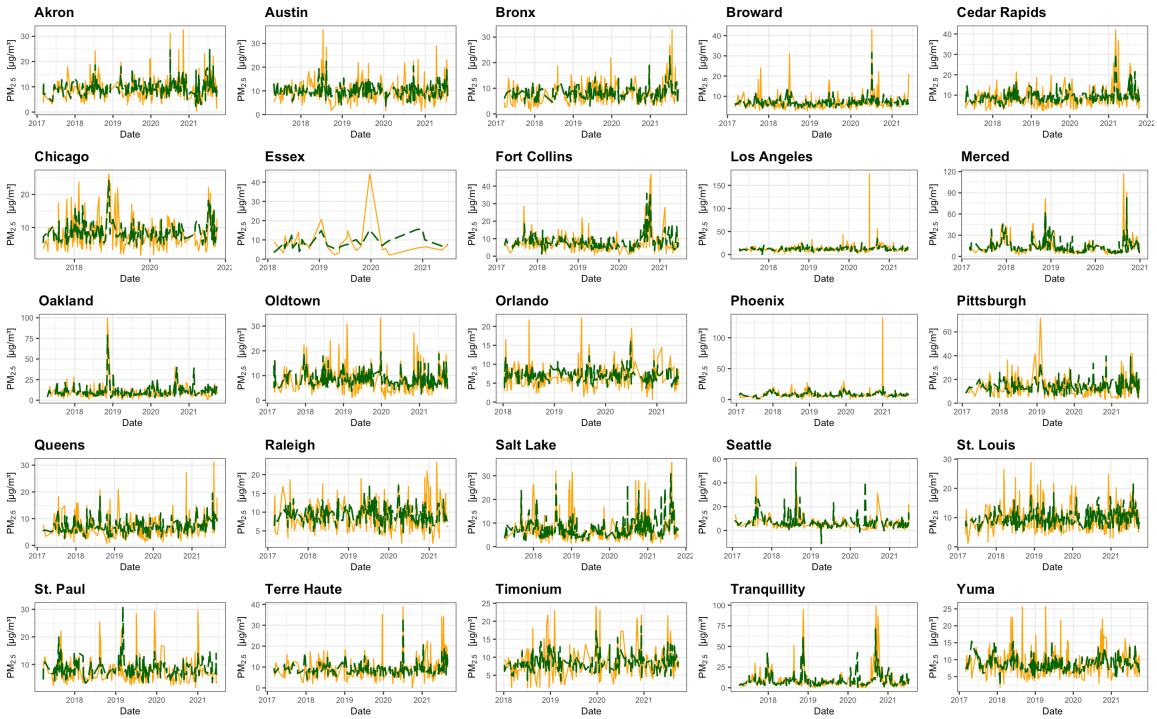


Figure 4-3: 25 Time Series Plots of XGBoost Predicted $PM_{2.5}$ against EPA $PM_{2.5}$. The orange solid line represents the actual EPA measurements, while the green dashed line represents the predictions from XGBoost model.

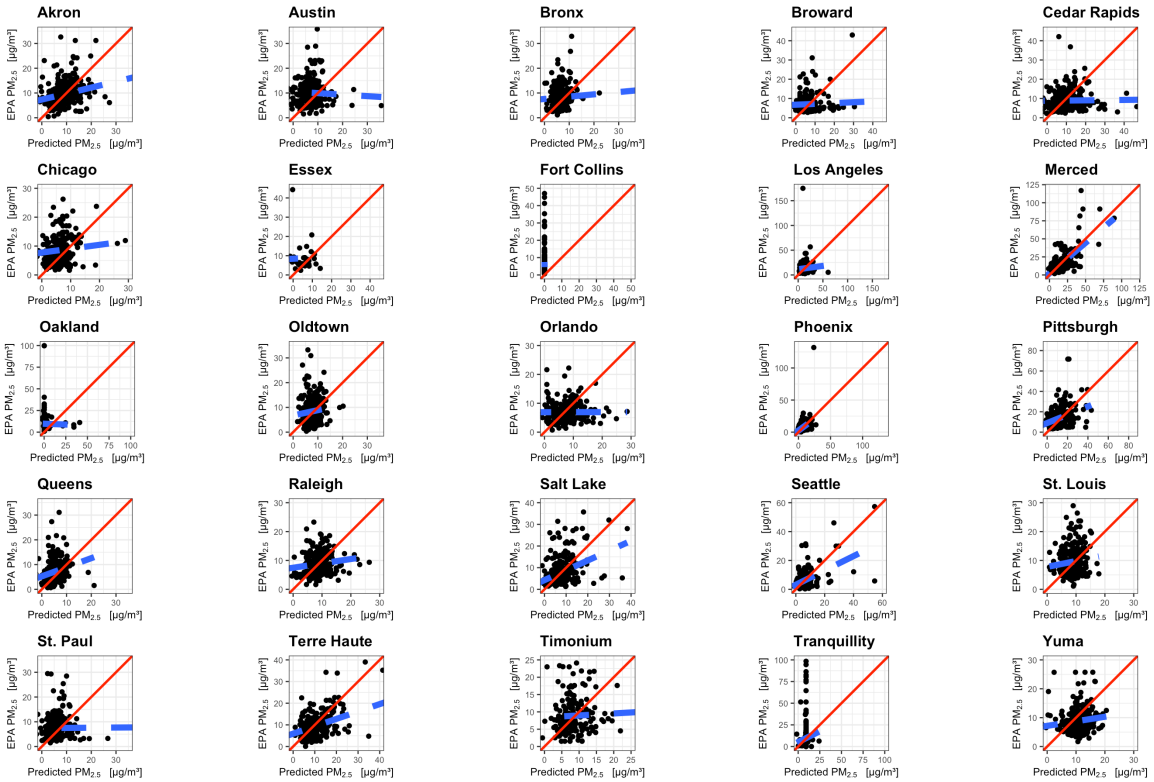


Figure 4-4: 25 Scatter Plots of CNN Predicted $PM_{2.5}$ against EPA $PM_{2.5}$. The black points are the data points in the testing set. The red solid line represents the 45° diagonal line, while the blue dashed line represents the fitted regression line.

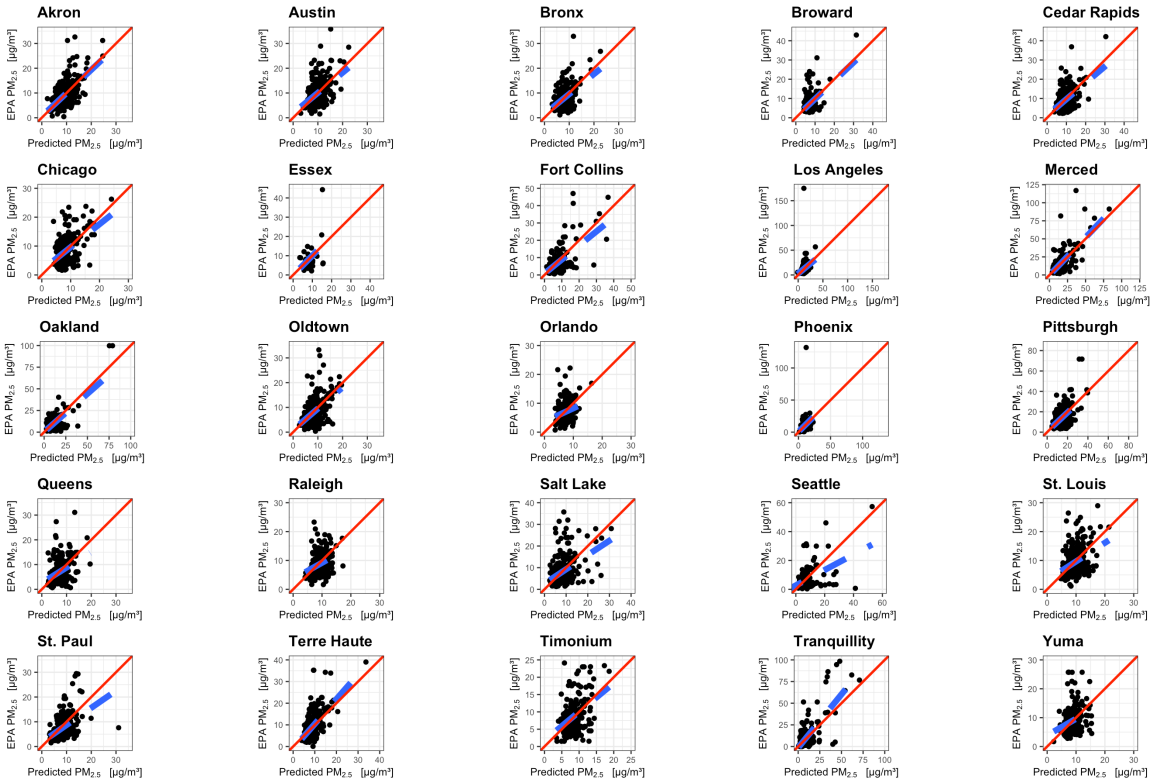


Figure 4-5: 25 Scatter Plots of XGboost Predicted $PM_{2.5}$ against EPA $PM_{2.5}$. The black points are the data points in the testing set. The red solid line represents the 45° diagonal line, while the blue dashed line represents the fitted regression line.

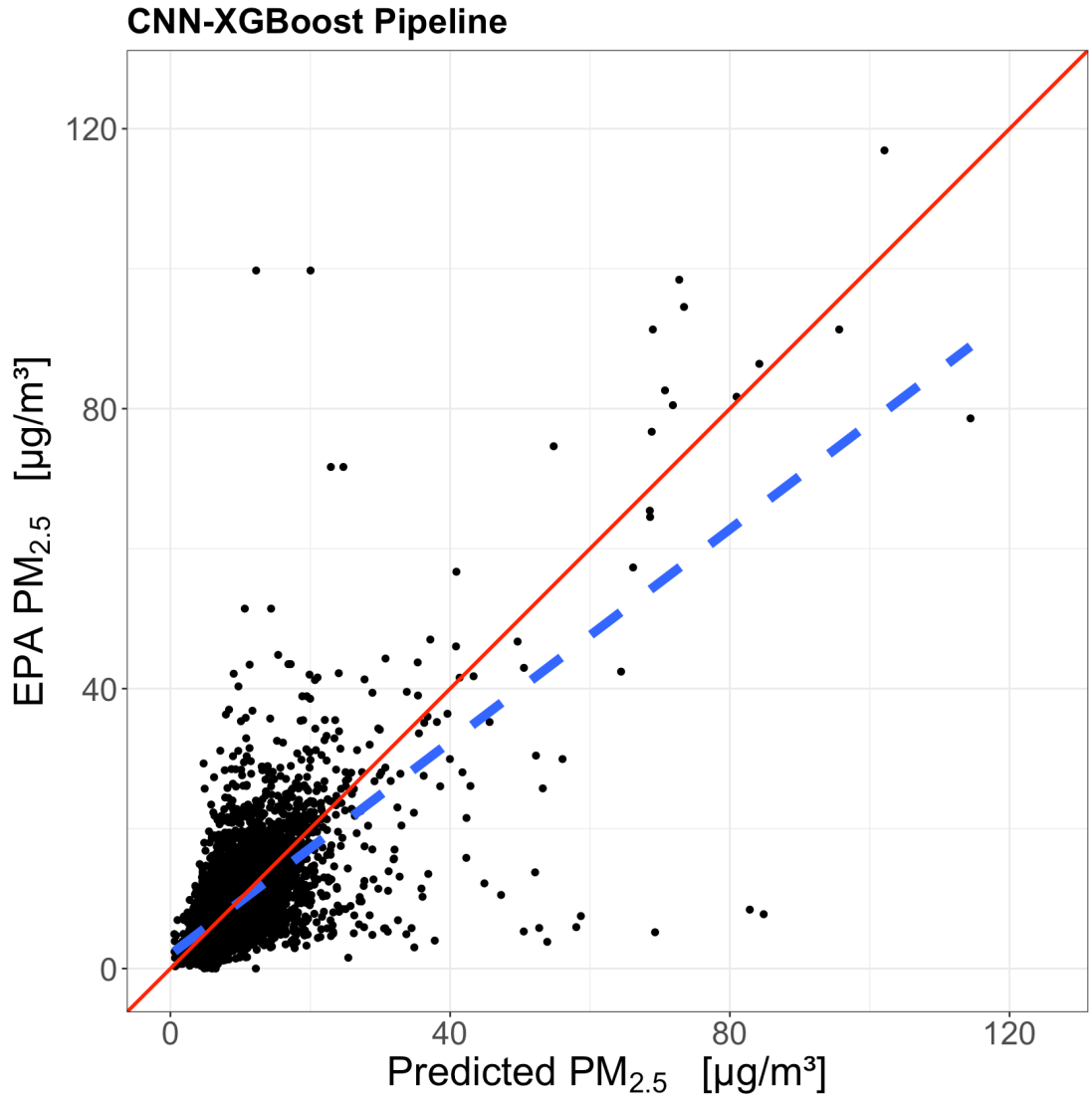


Figure 4-6: Scatter Plot of CNN-XGboost Predicted $PM_{2.5}$ against EPA $PM_{2.5}$. The black points are the data points in the testing set. The red solid line represents the 45° diagonal line, while the blue dashed line represents the fitted regression line.

Chapter 5

Discussion

5.1 Comparison with Related Studies

Three machine learning techniques, including XGBoost, CNN, and CNN-XGBoost pipeline, were used to predict $PM_{2.5}$ concentrations. The XGBoost technique demonstrated the highest performance and an acceptable time of training, with RMSE = 3.98 $\mu g\ m^{-3}$ and $R^2 = 0.65$. Within each location, the RMSE of XGBoost was lower than that of CNN, suggesting that XGBoost provided a more robust estimation. The CNN-XGBoost pipeline had an RMSE of 5.87 $\mu g\ m^{-3}$ and an R^2 of 0.37, and it tended to overestimate the actual $PM_{2.5}$ measurements. Even though XGBoost outperformed the other models, it was not able to capture some of the relatively high values and variability, leading to lower R^2. We compared our results with other studies.

Study	Data Source	Location	Duration (years)	$PM_{2.5}$ Mean (SD)	Model	Time-Scale	RMSE	R^2
This study	Satellite Images	25 locations in the U.S.	~ 5	8.94 (6.68) 9.13 (7.42)	XGBoost CNN-XGBoost	daily	3.98 5.87	0.65 0.37
Wang et al., 2017	Satellite-derived AOD	3 locations in China	~ 2		LME	daily	24.5	0.86
Zamani Joharestani et al., 2019	Satellite-derived AOD	Tehran, Iran	~ 4	86.8 (33)	RF XGBoost ANN	daily	14.47 13.62 14.56	0.78 0.80 0.77
Di, Kloog, et al., 2016	Satellite-derived AOD	49 locations in the U.S.	~ 13		CNN	daily	2.94	0.84
Datta et al., 2020	Low-Cost Sensors	Baltimore, U.S.	~ 0.7	8.0 (6.0)	MLR	daily	1.9	
Mukherjee et al., 2019	Low-Cost Sensors	Sacramento, U.S.	~ 0.08		MLR	daily		0.69
Johnson, Bonczak, and Kontokosta, 2018	Low-Cost Sensors	New York City, U.S.	~ 0.08	7.8	OLS RR GBRT	hourly	3.11 3.07 2.16	0.507 0.521 0.762
T. Zheng et al., 2020	Satellite Images	Beijing, China Shanghai, China	~ 3	42.7 (42.9) 38.4 (24.6)	VGG16-RF	daily	16.3 10.8	0.86 0.81

Table 5-1: Comparison of Model Performance with Other Studies. Abbreviations: Linear Mixed Effect (LME); Random forest (RF); Artificial Neural Network (ANN); Multiple Linear Regression (MLR); Ordinary Least Squares (OLS); Ridge Regression (RR); Gradient Boosting Regression Tree (GBRT); Visual Geometry Group that supports 16 layers (VGG16).

As seen in Table 5-1, the related works can be summarized to three main categories. In the first category, besides meteorological data, the satellite-derived data such as AOD products, were used to find a relationship between the AOD and $PM_{2.5}$ measurements through an appropriate model. In the second category, low-cost sensors and meteorological data were used to improve the spatial and temporal resolution of $PM_{2.5}$ through some calibration methods. The third category, including our study, used machine learning approaches to make predictions of $PM_{2.5}$ concentrations based on satellite images and meteorological data.

The most widely used method for $PM_{2.5}$ estimations from the satellite data is finding the relationship among the satellite-derived aerosol optical depth (AOD). Wang et al., 2017 used AOD through an LME, and the RMSE was 24.5 $\mu g\ m^{-3}$ and R^2 was 0.86. Even though the model performed well, their method was developed under clear-sky conditions and required filtering cloudy pixels. Zamani Joharestani et al., 2019 used AOD and conducted RF, ANN, and XGBoost models to predict $PM_{2.5}$ of Tehran's urban area in Iran. They used interpolation to estimate and fill in the missing data. In addition, they applied standard normalizations to reduce the instability during the model training. XGBoost outperformed with RMSE of 13.62 $\mu g\ m^{-3}$ and R^2 of 0.80, following by RF with RMSE of 14.47 $\mu g\ m^{-3}$ and R^2 of 0.78. ANN had a slightly higher RMSE of 14.56 $\mu g\ m^{-3}$ and a slightly lower R^2 of 0.77. Since the baseline $PM_{2.5}$ concentrations in Tehran were much higher, the RMSE's were higher than our study's. But the high R^2's suggested that their models had better fits. Some additional features used in this analysis, such as latitude, longitude, altitude, day of week, and season, were identified to be important variables and may be utilized in our study as well.

Di, Kloog, et al., 2016 conducted an analysis based on daily $PM_{2.5}$ concentrations of 48 contiguous states in the U.S. and Washington D.C. from January 2000 to December 2012. They set daily EPA measurements as reference, and included multiple input

variables such as AOD data, surface reflectance, chemical transport model outputs, meteorological data, aerosol index data, land-use terms, regional and monthly dummy variables, and scaling factor. In addition, they filled in any missing values by a neural network or a linear interpolation. A CNN model was used, where convolutional layers were implemented to account for the temporal and spatial autocorrelation. Their hybrid nationwide model achieved high performance with an average total R^2 of 0.84 (ranged from 0.74 to 0.88) and an average RMSE of 2.94 $\mu g\ m^{-3}$ (ranged from 2.64 $\mu g\ m^{-3}$ to 3.58 $\mu g\ m^{-3}$). In terms of season and region, the model performed the best in summer and in the Eastern U.S.. Various types of variables helped the model achieve higher prediction accuracy, yet limited its application to regions or countries with fewer data available.

Datta et al., 2020 deployed 45 low-cost sensors and utilized an MLR for on-field calibrated low-cost $PM_{2.5}$ networks in Baltimore (MD) from December 2018 to July 2019. The hourly RMSE was 3.6 $\mu g\ m^{-3}$ and daily RMSE was 1.9 $\mu g\ m^{-3}$. This approach provided well-calibrated measurements and was proven to be robust for the co-location monitor and the co-location season. Mukherjee et al., 2019 also used an MLR and deployed 19 low-cost sensors from December 2016 to January 2017 to determine the spatial variability of $PM_{2.5}$ in Sacramento (CA). During the study, the the highest daily correlation between the sensors and the regulatory monitors had a Pearson R^2 of 0.69. Johnson, Bonczak, and Kontokosta, 2018 carried out an analysis on low-cost sensors in New York City (NY) from February 2017 to March 2017. Different from our study, hourly measurements were usually collected and used for low-cost sensors networks. Among three modeling approaches (OLS, RR, and GBRT), GBRT had the most significant enhancement for relative calibration and large-scale deployments with hourly RMSE = 2.16 $\mu g\ m^{-3}$ and R^2 of 0.762. Overall, studies showed that low-cost sensors could be effective tools for $PM_{2.5}$ estimation with appropriate models and calibration methods.

T. Zheng et al., 2020 employed a VGG16 algorithm for feature extraction from satellite images of Beijing from 2017 to 2019, and the extracted image features were given to a RF as input to estimate the $PM_{2.5}$ concentrations. The RMSE of the best model was 16.3μg m^{-3} and R^2 was 0.86. To validate the models, they used the same pipeline to make predictions in Shanghai, where the RMSE was 10.8 μg m^{-3} and R^2 was 0.87. Their approach achieved quite high prediction accuracy since they restrained themselves to only the images on approximately uncloudy days (*i.e.* cloud coverages below 5%). They also resized all their images to 224 × 224 pixels and subtracted the mean RGB values of the original ImageNet training set from each pixel, which made the model more flexible in being adopted to other locations.

5.2 Limitations and Future Work

Some limitations remain in this study. Although XGBoost achieved reasonable overall $PM_{2.5}$ prediction performance, the RMSE varied from 2.95 μg m^{-3} to 4.73 μg m^{-3} and R^2 varied from 0.40 to 0.76 in different months, indicating some seasonal patterns were not fully captured. Also, the CNN architecture used in this study was not optimal, since some of the predictions were negative.

Future studies will explore the possibility of adding other components, such as seasons and day of the week, as covariates. Studies have shown that relatively high $PM_{2.5}$ concentrations tend to appear on Fridays and Saturdays, while on Mondays and Sundays the $PM_{2.5}$ concentrations are generally lower than those on most other days of the week (Zhao et al., 2018). To eliminate the negative predictions from CNN architecture, we could take the logarithm of the outcomes. Also, we will find a more appropriate approach for dealing with the missing pixels in the satellite imagery, such as a Random Forest imputation or interpolation. In addition, we will investigate the application of a CNN architecture using ResNet with 50 layers (K. He et al., 2016) or a VGG16 architecture (T. Zheng et al., 2020).

Chapter 6

Conclusion

In this study, we utilized XGBoost, CNN, and CNN-XGBoost pipeline to predict daily $PM_{2.5}$ concentrations in 25 locations across the U.S from January 2017 to October 2021. In designing our models, we included daily measurements of $PM_{2.5}$ from EPA monitors, satellite imagery and atmospheric information from Planet, and meteorological features from NASA POWER. Within each location, XGBoost provided more accurate estimates while CNN led to overestimation. Even though CNN resulted in lower prediction accuracy, it is flexible and is able to process the satellite imagery directly. It can detect the informative features and learn the dynamic changes without any human supervision. When we combined all observations from 25 locations, in comparison to CNN-XGBoost pipeline, XGBoost had a better performance of RMSE = 3.98 $\mu g\ m^{-3}$ and $R^2 = 0.65$. Since XGBoost is designed with faster training speed and lower memory usage, it takes less time than CNN to produce outputs. It is a highly effective and versatile method that is capable of handling large-scale data and providing more accurate predictions.

References

Akyüz, Mehmet and Hasan Çabuk (2009). "Meteorological variations of PM2. 5/PM10 concentrations and particle-associated polycyclic aromatic hydrocarbons in the atmospheric environment of Zonguldak, Turkey." In: *Journal of hazardous materials* 170.1, pp. 13–21.

Badura, Marek et al. (2018). "Evaluation of low-cost sensors for ambient PM2. 5 monitoring." In: *Journal of Sensors* 2018.

Borghi, Francesca et al. (2021). "Estimation of the Inhaled Dose of Pollutants in Different Micro-Environments: A Systematic Review of the Literature." In: *Toxics* 9.6, p. 140.

Brekke, Camilla and Anne HS Solberg (2005). "Oil spill detection by satellite remote sensing." In: *Remote sensing of environment* 95.1, pp. 1–13.

Chen, Tianqi and Carlos Guestrin (2016). "Xgboost: A scalable tree boosting system." In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Chen, Tianqi, Tong He, et al. (2015). "Xgboost: extreme gradient boosting." In: *R package version 0.4-2* 1.4, pp. 1–4.

Chow, Judith C (1995). "Measurement methods to determine compliance with ambient air quality standards for suspended particles." In: *Journal of the Air & Waste Management Association* 45.5, pp. 320–382.

Datta, Abhirup et al. (2020). "Statistical field calibration of a low-cost PM2. 5 monitoring network in Baltimore." In: *Atmospheric Environment* 242, p. 117761.

Di, Qian, Lingzhen Dai, et al. (2017). "Association of short-term exposure to air pollution with mortality in older adults." In: *Jama* 318.24, pp. 2446–2456.

Di, Qian, Itai Kloog, et al. (2016). "Assessing PM2. 5 exposures with high spatiotemporal resolution across the continental United States." In: *Environmental science & technology* 50.9, pp. 4712–4721.

Fan, Jingchun et al. (2016). "The impact of $PM_{2.5}$ on asthma emergency department visits: a systematic review and meta-analysis." In: *Environmental Science and Pollution Research* 23.1, pp. 843–850.

Gao, Meiling, Junji Cao, and Edmund Seto (2015). "A distributed network of low-cost continuous reading sensors to measure spatiotemporal variations of PM2. 5 in Xi'an, China." In: *Environmental pollution* 199, pp. 56–65.

Guo, Bin et al. (2021). "Estimating PM2. 5 concentrations via random forest method using satellite, auxiliary, and ground-level station dataset at multiple temporal scales across China in 2017." In: *Science of The Total Environment* 778, p. 146288.

Hamanaka, Robert B and Gökhan M Mutlu (2018). "Particulate matter air pollution: effects on the cardiovascular system." In: *Frontiers in endocrinology* 9, p. 680.

Hayes, Richard B et al. (2020). "$PM_{2.5}$ air pollution and cause-specific cardiovascular disease mortality." In: *International journal of epidemiology* 49.1, pp. 25–35.

He, Kaiming et al. (2016). "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Holloway, Jacinta and Kerrie Mengersen (2018). "Statistical machine learning methods and remote sensing for sustainable development goals: a review." In: *Remote Sensing* 10.9, p. 1365.

Holmgren, Peter and Thomas Thuresson (1998). "Satellite remote sensing for forestry planning—a review." In: *Scandinavian Journal of Forest Research* 13.1-4, pp. 90–110.

Hooper, Laura G et al. (2018). "Ambient air pollution and chronic bronchitis in a cohort of US women." In: *Environmental health perspectives* 126.2, p. 027005.

Johnson, Nicholas E, Bartosz Bonczak, and Constantine E Kontokosta (2018). "Using a gradient boosting model to improve the performance of low-cost aerosol monitors in a dense, heterogeneous urban environment." In: *Atmospheric environment* 184, pp. 9–16.

Just, Allan C et al. (2018). "Correcting measurement error in satellite aerosol optical depth with machine learning for modeling PM2. 5 in the Northeastern USA." In: *Remote sensing* 10.5, p. 803.

Karandana Gamalathge, TD and M Green (2017). "How Winter Time Atmospheric Stability Influences PM2. 5 Concentration in Different Complex Terrains; Beijing in China vs Fairbanks in Alaska." In: *AGU Fall Meeting Abstracts*. Vol. 2017, A53B–2239.

Keet, Corinne A, Joshua P Keller, and Roger D Peng (2018). "Long-term coarse particulate matter exposure is associated with asthma among children in Medicaid." In: *American journal of respiratory and critical care medicine* 197.6, pp. 737–746.

Kleine Deters, Jan et al. (2017). "Modeling PM2. 5 urban pollution using machine learning and selected meteorological parameters." In: *Journal of Electrical and Computer Engineering* 2017.

Kumar, Prashant et al. (2015). "The rise of low-cost sensing for managing air pollution in cities." In: *Environment international* 75, pp. 199–205.

Li, Zheng et al. (2019). "Longitudinal effect of ambient air pollution and pollen exposure on asthma control: the Patient-Reported Outcomes Measurement Information System (PROMIS) pediatric asthma study." In: *Academic Pediatrics* 19.6, pp. 615–623.

Lin, Changqing et al. (2015). "Using satellite remote sensing data to estimate the high-resolution distribution of ground-level PM2. 5." In: *Remote Sensing of Environment* 156, pp. 117–128.

Lipsett, Michael J et al. (2011). "Long-term exposure to air pollution and cardiorespiratory disease in the California teachers study cohort." In: *American journal of respiratory and critical care medicine* 184.7, pp. 828–835.

Liu, Wei et al. (2019). "Meteorological pattern analysis assisted daily PM2. 5 grades prediction using SVM optimized by PSO algorithm." In: *Atmospheric Pollution Research* 10.5, pp. 1482–1491.

Liu, Yang et al. (2005). "Estimating ground-level PM2. 5 in the eastern United States using satellite remote sensing." In: *Environmental science & technology* 39.9, pp. 3269–3278.

Ma, Jinghui et al. (2020). "Application of the XGBoost machine learning method in PM2. 5 prediction: A case study of Shanghai." In: *Aerosol and Air Quality Research* 20.1, pp. 128–138.

Manisalidis, Ioannis et al. (2020). "Environmental and health impacts of air pollution: a review." In: *Frontiers in public health*, p. 14.

Martin, Randall V (2008). "Satellite remote sensing of surface air quality." In: *Atmospheric environment* 42.34, pp. 7823–7843.

Morawska, Lidia et al. (2018). "Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone?" In: *Environment international* 116, pp. 286–299.

Mukherjee, Anondo et al. (2019). "Measuring spatial and temporal PM2. 5 variations in Sacramento, California, communities using a network of low-cost sensors." In: *Sensors* 19.21, p. 4701.

Murray, Christopher JL et al. (2020). "Global burden of 87 risk factors in 204 countries and territories, 1990–2019: a systematic analysis for the Global Burden of Disease Study 2019." In: *The Lancet* 396.10258, pp. 1223–1249.

Muthukumar, Pratyush et al. (2021). "Predicting $PM_{2.5}$ atmospheric air pollution using deep learning with meteorological data and ground-based observations and remote-sensing satellite big data." In: *Air Quality, Atmosphere & Health*, pp. 1–14.

O'Shea, Keiron and Ryan Nash (2015). "An introduction to convolutional neural networks." In: *arXiv preprint arXiv:1511.08458*.

O'Donnell, Martin J et al. (2011). "Fine particulate air pollution ($PM_{2.5}$) and the risk of acute ischemic stroke." In: *Epidemiology (Cambridge, Mass.)* 22.3, p. 422.

Ozesmi, Stacy L and Marvin E Bauer (2002). "Satellite remote sensing of wetlands." In: *Wetlands ecology and management* 10.5, pp. 381–402.

Pak, Unjin et al. (2020). "Deep learning-based PM2. 5 prediction considering the spatiotemporal correlations: A case study of Beijing, China." In: *Science of The Total Environment* 699, p. 133561.

Parcak, Sarah H (2009). *Satellite remote sensing for archaeology*. Routledge.

Piedrahita, Ricardo et al. (2014). "The next generation of low-cost personal air quality sensors for quantitative exposure monitoring." In: *Atmospheric Measurement Techniques* 7.10, pp. 3325–3336.

Roberts, Susanna et al. (2019). "Exploration of NO2 and $PM_{2.5}$ air pollution and mental health problems using high-resolution data in London-based children from a UK longitudinal cohort study." In: *Psychiatry research* 272, pp. 8–17.

Stowell, Jennifer D et al. (2020). "Estimating PM2. 5 in Southern California using satellite data: factors that affect model performance." In: *Environmental Research Letters* 15.9, p. 094004.

Sun, Jin, Jianhua Gong, and Jieping Zhou (2021). "Estimating hourly PM2. 5 concentrations in Beijing with satellite aerosol optical depth and a random forest approach." In: *Science of The Total Environment* 762, p. 144502.

Van Donkelaar, Aaron, Randall V Martin, and Rokjin J Park (2006). "Estimating ground-level PM2. 5 using aerosol optical depth determined from satellite remote sensing." In: *Journal of Geophysical Research: Atmospheres* 111.D21.

Wang, Wei et al. (2017). "Deriving hourly PM2. 5 concentrations from himawari-8 aods over beijing–tianjin–hebei in China." In: *Remote Sensing* 9.8, p. 858.

Xing, Yu-Fei et al. (2016). "The impact of $PM_{2.5}$ on the human respiratory system." In: *Journal of thoracic disease* 8.1, E69.

Yang, Jun et al. (2013). "The role of satellite remote sensing in climate change studies." In: *Nature climate change* 3.10, pp. 875–883.

Yanosky, Jeff D et al. (2014). "Spatio-temporal modeling of particulate air pollution in the conterminous United States using geographic and meteorological predictors." In: *Environmental Health* 13.1, pp. 1–15.

Zamani Joharestani, Mehdi et al. (2019). "PM2. 5 prediction based on random forest, XGBoost, and deep learning using multisource remote sensing data." In: *Atmosphere* 10.7, p. 373.

Zhao, Naizhuo et al. (2018). "Day-of-week and seasonal patterns of PM2. 5 concentrations over the United States: Time-series analyses using the Prophet procedure." In: *Atmospheric environment* 192, pp. 116–127.

Zheng, Tongshu et al. (2020). "Estimating ground-level $PM_{2.5}$ using micro-satellite images by a convolutional neural network and random forest approach." In: *Atmospheric Environment* 230, p. 117451.

Zhou, Xiaodan et al. (2021). "Excess of COVID-19 cases and deaths due to fine particulate matter exposure during the 2020 wildfires in the United States." In: *Science Advances* 7.33, eabi8789.