



AIR QUALITY PREDICTION USING A HYBRID DEEP LEARNING ARCHITECTURE

AYŞENUR GİLİK

MASTER'S THESIS

Submitted to the School of Graduate Studies of
Kadir Has University in partial fulfillment of the requirements for the degree of
Master of Science in Electronics Engineering Graduate Program

İSTANBUL, AUGUST, 2020

DECLARATION OF RESEARCH ETHICS /
METHODS OF DISSEMINATION

I, AYŞENUR GİLİK, hereby declare that;

- this master's thesis is my own original work and that due references have been appropriately provided on all supporting literature and resources;
- this master's thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- I have followed *Kadir Has University Academic Ethics Principles prepared in accordance with The Council of Higher Education's Ethical Conduct Principles*.

In addition, I understand that any false claim in respect of this work will result in disciplinary action in accordance with University regulations.

Furthermore, both printed and electronic copies of my work will be kept in Kadir Has Information Center under the following condition as indicated below

The full content of my thesis will be accessible from everywhere by all means.

AYŞENUR GİLİK

05.08.2020

KADİR HAS UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ACCEPTANCE AND APPROVAL

This work entitled AIR QUALITY PREDICTION USING A HYBRID DEEP LEARNING ARCHITECTURE prepared by AYŞENUR GİLİK has been judged to be successful at the defense exam on 05.08.2020 and accepted by our jury as

APPROVED BY:

Assoc. Prof. Dr. Atilla Özmen (Advisor)
Kadir Has University

Asst. Prof. Dr. Arif Selçuk Öğrenci (Co-advisor)
Kadir Has University

Asst. Prof. Dr. Baran Tander
Kadir Has University

Asst. Prof. Dr. Yalçın Şadi
Kadir Has University

Asst. Prof. Dr. Figen Özen
Haliç University

I certify that the above signatures belong to the faculty members named above.

.....

Prof. Dr. Emine Füsun Alioğlu

Dean of School of Graduate Studies

DATE OF APPROVAL: 05.08.2020

TABLE OF CONTENTS

ABSTRACT	i
ÖZET	ii
ACKNOWLEDGEMENTS	iii
DEDICATION	iv
LIST OF TABLES	v
LIST OF FIGURES	vi
1. INTRODUCTION	1
1.1 Objective	1
1.2 Motivation	2
1.3 Subject and Scope	6
1.4 Contribution and Impact	11
2. LITERATURE	14
3. METHODOLOGY	20
3.1 Data	23
3.2 Time Series Data Set Preparation	24
3.2.1 Rolling-Window	24
3.2.2 Input Structure of CNN	25
3.3 Artificial Neural Networks	25
3.3.1 Convolutional Neural Network (CNN)	30
3.3.2 Recurrent Neural Networks	32
3.3.3 Long Short-Term Memory	34
3.4 Transfer Learning	36
3.5 Performance Metrics	39
4. IMPLEMENTATION	41
4.1 Data Preprocessing	44
4.1.1 Linear Interpolation	44
4.1.2 Data Normalization	47
4.2 Hyperparameter Tuning	48

5. RESULTS	51
5.1 Method-1	53
5.2 Method-2	56
5.3 Method-3	58
5.4 Transfer Learning	61
6. CONCLUSION	64
APPENDIX A: Descriptive Statistics	76
APPENDIX B: Histogram for the Concentration of Pollutants in Cities	
78	
APPENDIX C: Time Series Graphs	82
APPENDIX D: Results of Barcelona	86
APPENDIX E: Results of Kocaeli	96
APPENDIX F: Results of İstanbul Dataset-1	99
APPENDIX G: Results of İstanbul Dataset-2	102
APPENDIX H: Results with Meteorological Data	105
H.1 Results for İstanbul Dataset-2	105
H.2 Results for Kocaeli	108
REFERENCES	112

AIR QUALITY PREDICTION USING A HYBRID DEEP LEARNING ARCHITECTURE

ABSTRACT

Air pollution prediction is related to the variables in environmental monitoring data and modeling of the complex relationship between these variables. The objectives of the thesis are to develop a supervised model for the prediction of air pollution by using real sensor data and to transfer the model between cities. A CNN+LSTM deep neural network model was developed to predict the concentration of air pollutants in multiple locations by using a spatial-temporal relationship. The 2D input (univariate) contains the information of one pollutant; the 3D input (multivariate) contains the information of all pollutants and meteorology. There are three methods employed according to the input-output type: Method-1 is based on univariate-input and univariate-output; Method-2 is based on multivariate input and univariate-output; Method-3 is based on multivariate input and multivariate output. The study was carried out for different pollutants which are in publicly available data of the cities of Barcelona, Kocaeli, and İstanbul. The hyperparameters were tuned to determine the architecture that achieved the lowest test RMSE. Comparing the performance of the CNN+LSTM network with a 1-hidden layer LSTM network, the proposed model improved the prediction performance by the rates between 11%-53% for PM_{10} , 20%-31% for O_3 , 9%-47% for NO_X and 18%-46% for SO_2 . After, the network weights were transferred from the source domains to the target domain. The model has a more reliable prediction performance with the transfer of the network from Kocaeli to İstanbul because of the similarities between those two cities.

Keywords: Deep learning, air pollution, prediction, CNN, LSTM, transfer learning.

HİBRİT DERİN ÖĞRENME MİMARİSİ KULLANARAK HAVA KALİTESİ TAHMİNİ

ÖZET

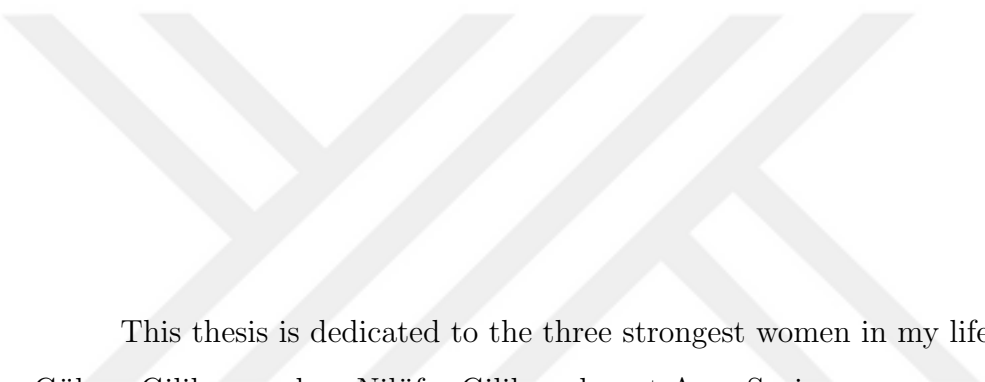
Hava kirliliği tahmini, çevresel izleme verilerindeki değişkenlerle ve bu değişkenler arasındaki karmaşık ilişkinin modellenmesiyle ilgilidir. Tezin amacı, gerçek sensör verilerini kullanarak hava kirliliğinin tahmini için denetimli bir model geliştirmek ve modeli şehirler arasında aktarmaktır. Mekansal-zamansal bir ilişki kullanarak birden çok konumdaki hava kirlleticilerinin konsantrasyonunu tahmin etmek için bir CNN+LSTM derin sinir ağı modeli geliştirildi. 2B ve 3B girdi yapıları tanımlanır: 2D girdi (tek değişkenli) bir kirleticinin bilgisini içerir; 3D girdi (çok değişkenli) tüm kirleticiler ve meteoroloji bilgilerini içerir. Girdi-çıkıya göre üç farklı yöntem vardır: Yöntem-1 tek değişkenli girdi ve tek değişkenli çıkıya dayanır; Yöntem-2, çok değişkenli girdi ve tek değişkenli çıkıya dayanmaktadır; Yöntem-3, çok değişkenli girdi ve çok değişkenli çıkıya dayanmaktadır. Çalışma Barselona, Kocaeli ve İstanbul şehirlerinin kamuya açık verilerinde bulunan farklı kirleticiler için yapılmıştır. Hiperparametreler, en düşük RMSE testine ulaşan mimariyi belirlemek için ayarlandı. CNN+LSTM ağının performansı 1 gizli katmanlı LSTM ağıyla karşılaştırıldığında, önerilen model tahmin performansını PM_{10} için 11%-53%, O_3 için 20%-31%, NO_X için 9%-47% ve SO_2 için 18%-46% arasındaki oranlarla geliştirdi. Ardından ağ ağırlıkları kaynak alanlardan hedef alana aktarıldı. Model, ağın Kocaeli'den İstanbul'a aktarımı ile bu iki şehir arasındaki benzerliklerden dolayı daha güvenilir bir tahmin performansına sahiptir.

Anahtar Sözcükler: Derin öğrenme, hava kirliliği, tahmin, CNN, LSTM, transfer öğrenimi.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisors Assoc. Prof. Dr. Atilla Özmen and Asst. Prof. Dr. Arif Selçuk Öğrenci for the invaluable contribution, support, and guidance through the learning process of this master thesis. They provided me with encouragement and patience throughout the duration of my education, and the door to their offices was always open whenever I had a question or I ran into a trouble spot. Furthermore, I must also thank Asst. Prof. Dr. Figen Özen who always supported me about my study field with her constructive advice.





This thesis is dedicated to the three strongest women in my life; to my mom Gülnur Gilik, grandma Nilüfer Gilik and aunt Ayşe Sevim.

LIST OF TABLES

Table 2.1	Proposed architectures and their performances in literature . . .	18
Table 3.1	Explanation of the methods according to the input-output type .	21
Table 3.2	Explanation of the data properties used for the methods	22
Table 5.1	Proposed deep learning model’s layers and their properties	52
Table 5.2	Test RMSE results for Kocaeli and İstanbul with air quality (AQ) and meteorological information (MI) data	61
Table 6.1	Explanation of the methods used in the implementation	65
Table 6.2	Optimal hyperparameter values for the air pollution prediction model	66
Table 6.3	Explanation of the methods and the data used for different cities	67
Table 6.4	Performance metrics for prediction results of Barcelona data . . .	69
Table 6.5	Performance metrics for prediction results of Kocaeli data	69
Table 6.6	Performance metrics for prediction results of İstanbul data	70
Table 6.7	Data properties and methods that give the best prediction results	71
Table 6.8	Comparison of the prediction performance between 1-hidden layer LSTM model and CNN+LSTM model in Barcelona	72
Table 6.9	Comparison of the prediction performance between 1-hidden layer LSTM model and CNN+LSTM model in Kocaeli	72
Table 6.10	Comparison of the prediction performance between 1-hidden layer LSTM model and CNN+LSTM model in İstanbul	73
Table A.1	Descriptive sStatistics for Barcelona Data (before data prepro- cessing)	76
Table A.2	Descriptive sStatistics for Kocaeli Data (before data preprocessing)	77
Table A.3	Descriptive Statistics for İstanbul Data (before data preprocessing)	77

LIST OF FIGURES

Figure 3.1	Rolling window method for the time series data	24
Figure 3.2	$m \times n$ input matrix for Convolutional Neural Network where m is the number of locations and n is the number of time step.	25
Figure 3.3	$m \times n \times k$ input matrix for Convolutional Neural Network	26
Figure 3.4	Working principle of a neuron	27
Figure 3.5	Feedforward Neural Network structure	27
Figure 3.6	CNN+LSTM Deep Learning structure	30
a	The graph of the sigmoid activation function	32
b	The graph of the ReLU activation function	32
Figure 3.8	Two nonlinear activation functions used in neural networks	32
Figure 3.9	Recurrent Neural Network structure	33
Figure 3.10	An RNN cell model	35
Figure 3.11	The RNN cell model with memory unit	35
Figure 3.12	The RNN cell model with memory and selection units	36
Figure 3.13	An LSTM cell model	36
Figure 3.14	Representation of transfer learning between two neural network models	39
Figure 4.1	Flowchart of proposed air pollution prediction system	43
Figure 4.2	Histogram for the concentration of O_3 in Barcelona	44
Figure 4.3	Histogram for the concentration of NO_2 in Barcelona	45
Figure 4.4	Histogram for the concentration of PM_{10} in Barcelona	45
Figure 4.5	Hourly concentration of O_3 in Barcelona	46
Figure 4.6	Hourly concentration of NO_2 in Barcelona	46
Figure 4.7	Hourly concentration of PM_{10} in Barcelona	47
Figure 4.8	Consecutive and sequential sample selection types	50
Figure 5.1	Results of Method-1 for the prediction of PM_{10} concentration in Barcelona (with non-normalized air pollution data)	54

Figure 5.2	Graph of the predicted and target values for the prediction of PM_{10} concentration in Barcelona (with non-normalized air pollution data) by Method-1	54
Figure 5.3	Results of Method-1 for the prediction of PM_{10} concentration in Barcelona (with normalized air pollution data)	55
Figure 5.4	Graph of the predicted and target values for the prediction of PM_{10} concentration in Barcelona (with normalized air pollution data) by Method-1	55
Figure 5.5	Scatter plot of target and predicted concentrations for PM_{10} in Barcelona (with air pollution data) by Method-2 and Method-3 .	56
Figure 5.6	Graph of the predicted and target values for the prediction of PM_{10} concentration in Barcelona (with normalized air pollution data) by Method-2 and Method-3	57
Figure 5.9	Scatter plot of target and predicted concentrations for PM_{10} in İstanbul (with air pollution data) by Method-2 and Method-3 (Dataset-1)	57
Figure 5.7	Scatter plot of target and predicted concentrations for PM_{10} in Kocaeli (with air pollution data) by Method-2 and Method-3 . .	58
Figure 5.10	Graph of the predicted and target values for the prediction of PM_{10} concentration in İstanbul (with Dataset-1) by Method-2 and Method-3	58
Figure 5.8	Graph of the predicted and target values for the prediction of PM_{10} concentration in Kocaeli (with normalized air pollution data) by Method-2 and Method-3	59
Figure 5.11	Scatter plot of target and predicted concentrations for PM_{10} in İstanbul (with air pollution data) by Method-2 and Method-3 (Dataset-2)	60
Figure 5.12	Graph of the predicted and target values for the prediction of PM_{10} concentration in İstanbul (with Dataset-2) by Method-2 and Method-3	60

Figure 5.13	Scatter plot for PM_{10} concentration in İstanbul by Method-2 with transfer learning, pre-trained neural network with Kocaeli data are tested on İstanbul data.	63
Figure 5.14	Scatter plot for PM_{10} concentration in İstanbul by Method-3 with transfer learning, pre-trained neural network with Kocaeli data are tested on İstanbul data.	63
Figure B.1	Histogram for the concentration of PM_{10} in Kocaeli	78
Figure B.2	Histogram for the concentration of SO_2 in Kocaeli	78
Figure B.3	Histogram for the concentration of NO_X in Kocaeli	79
Figure B.4	Histogram for the concentration of O_3 in İstanbul	79
Figure B.5	Histogram for the concentration of NO_2 in İstanbul	80
Figure B.6	Histogram for the concentration of PM_{10} in İstanbul	80
Figure B.7	Histogram for the concentration of SO_2 in İstanbul	81
Figure B.8	Histogram for the concentration of NO_X in İstanbul	81
Figure C.1	Hourly concentration of PM_{10} in Kocaeli	82
Figure C.2	Hourly concentration of SO_2 in Kocaeli	82
Figure C.3	Hourly concentration of NO_X in Kocaeli	83
Figure C.4	Hourly concentration of O_3 in İstanbul	83
Figure C.5	Hourly concentration of NO_2 in İstanbul	84
Figure C.6	Hourly concentration of PM_{10} in İstanbul	84
Figure C.7	Hourly concentration of SO_2 in İstanbul	85
Figure C.8	Hourly concentration of NO_X in İstanbul	85
Figure D.1	Scatter plot of Method-1 test results for O_3 in Barcelona with non-normalized data	86
Figure D.2	Scatter plot of Method-1 test results for NO_2 in Barcelona with non-normalized data	87
Figure D.3	Scatter plot of Method-1 test results for PM_{10} in Barcelona with non-normalized data	88
Figure D.4	Graph of the target-predicted values of Method-1 test results for O_3 in Barcelona with non-normalized data	88

Figure D.5	Graph of the target-predicted values of Method-1 test results for NO_2 in Barcelona with non-normalized data	89
Figure D.6	Graph of the target-predicted values of Method-1 test results for PM_{10} in Barcelona with non-normalized data	89
Figure D.7	Scatter plot of Method-1 test results for O_3 in Barcelona with normalized data	90
Figure D.8	Scatter plot of Method-1 test results for NO_2 in Barcelona with normalized data	90
Figure D.9	Scatter plot of Method-1 test results for PM_{10} in Barcelona with normalized data	91
Figure D.10	Graphs of the target-predicted values of Method-1 test results for O_3 in Barcelona with normalized data	91
Figure D.11	Graphs of the target-predicted values of Method-1 test results for NO_2 in Barcelona with normalized data	92
Figure D.12	Graphs of the target-predicted values of Method-1 test results for PM_{10} in Barcelona with normalized data	92
Figure D.13	Scatter plot of Method-2 and Method-3 test results for O_3 in Barcelona with normalized data	93
Figure D.14	Scatter plot of Method-2 and Method-3 test results for NO_2 in Barcelona with normalized data	93
Figure D.15	Scatter plot of Method-2 and Method-3 test results for PM_{10} in Barcelona with normalized data	94
Figure D.16	Graphs of the target-predicted values of Method-2 and Method-3 test results for O_3 in Barcelona with normalized data	94
Figure D.17	Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_2 in Barcelona with normalized data	95
Figure D.18	Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in Barcelona with normalized data	95
Figure E.1	Scatter plot of Method-2 and Method-3 test results for PM_{10} in Kocaeli with normalized data	96

Figure E.2	Scatter plot of Method-2 and Method-3 test results for SO_2 in Kocaeli with normalized data	96
Figure E.3	Scatter plot of Method-2 and Method-3 test results for NO_X in Kocaeli with normalized data	97
Figure E.4	Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in Kocaeli with normalized data	97
Figure E.5	Graphs of the target-predicted values of Method-2 and Method-3 test results for SO_2 in Kocaeli with normalized data	98
Figure E.6	Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_X in Kocaeli with normalized data	98
Figure F.1	Scatter plot of Method-2 and Method-3 test results for O_3 in İstanbul (Dataset-1) with normalized data	99
Figure F.2	Scatter plot of Method-2 and Method-3 test results for NO_2 in İstanbul (Dataset-1) with normalized data	99
Figure F.3	Scatter plot of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-1) with normalized data	100
Figure F.4	Graphs of the target-predicted values of Method-2 and Method-3 test results for O_3 in İstanbul (Dataset-1) with normalized data .	100
Figure F.5	Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_2 in İstanbul (Dataset-1) with normalized data	101
Figure F.6	Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-1) with normalized data	101
Figure G.1	Scatter plot of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-2) with normalized data	102
Figure G.2	Scatter plot of Method-2 and Method-3 test results for SO_2 in İstanbul (Dataset-2) with normalized data	102
Figure G.3	Scatter plot of Method-2 and Method-3 test results for NO_X in İstanbul (Dataset-2) with normalized data	103

Figure G.4	Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-2) with normalized data	103
Figure G.5	Graphs of the target-predicted values of Method-2 and Method-3 test results for SO_2 in İstanbul (Dataset-2) with normalized data	104
Figure G.6	Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_X in İstanbul (Dataset-2) with normalized data	104
Figure H.1	Scatter plot of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-2) with meteorological data	105
Figure H.2	Scatter plot of Method-2 and Method-3 test results for SO_2 in İstanbul (Dataset-2) with meteorological data	105
Figure H.3	Scatter plot of Method-2 and Method-3 test results for NO_X in İstanbul (Dataset-2) with meteorological data	106
Figure H.4	Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-2) with meteorological data	106
Figure H.5	Graphs of the target-predicted values of Method-2 and Method-3 test results for SO_2 in İstanbul (Dataset-2) with meteorological data	107
Figure H.6	Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_X in İstanbul (Dataset-2) with meteorological data	107
Figure H.7	Scatter plot of Method-2 and Method-3 test results for PM_{10} in Kocaeli with meteorological data	108
Figure H.8	Scatter plot of Method-2 and Method-3 test results for SO_2 in Kocaeli with meteorological data	108
Figure H.9	Scatter plot of Method-2 and Method-3 test results for NO_X in Kocaeli with meteorological data	109

Figure H.10	Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in Kocaeli with meteorological data	109
Figure H.11	Graphs of the target-predicted values of Method-2 and Method-3 test results for SO_2 in Kocaeli with meteorological data	110
Figure H.12	Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_X in Kocaeli with meteorological data	110
Figure H.13	Graphs of the target-predicted values for PM_{10} in İstanbul (Dataset-2) with transfer learning	111



1. INTRODUCTION

1.1 Objective

We have two sets of research questions in this thesis.

- i. How successful can we develop a supervised model for the prediction of air pollution by using real sensor data obtained in different locations of a city? What are the architectures and set of parameters that give an optimal level of learning? How do those parameters change among different pollutants?
- ii. How successful can we transfer a model obtained by training data of a city/region to a different city? Are the parameters that give an optimal level of learning the same for different cities?

The objective of the research is to develop a solution that can predict the hourly concentration of the air pollutants (such as Nitrogen Oxides (NO_X), Sulphur Dioxide (SO_2), Ozone (O_3), and Particulate Matter (PM)) as indicator of air pollution with or without use of meteorological information to obtain a smarter city environment.

The target of the research is to develop an algorithm

- to preprocess sensor data
- to fuse data from multiple sensor types and locations
- to determine the optimal deep learning model and hyperparameters for time series prediction
- to predict air pollutants' concentration using machine learning algorithms.

A model for the prediction will be developed based on past data that is a time

series. The model will be developed separately for three cities (Barcelona, İstanbul, Kocaeli) for which open data are available. An outcome of the research will be a comparison of the sets of features for optimal performance.

The second target is to investigate the transferability of the model developed for a city to other cities. Factors such as which features achieve optimal prediction performance and which trained model has prediction ability on weight/feature transfer of the target city are examined. The outcome will be a low-cost algorithm to predict hourly air pollution and perform feature/weight transfer for multiple cities.

1.2 Motivation

Since air pollution is one of the significant threats to the health of society living in cities worldwide, governments have considered clean air as a basic requirement of health and well-being. WHO (World Health Organization) assessment points out that more than 3 million premature deaths per year are caused by urban outdoor and indoor air pollution [1]. Some air pollutants like O_3 and Nitrogen Oxides (NO_X) can affect ecosystems and vegetation directly, so the quality of water and soil that support the ecosystem get damaged [2]. Additionally, air pollution has negative effects on the economy; a decrease in the lifetime of individuals, an increase in medical cost, reducing productivity. European Commission offers the 2030 climate and energy framework [3]. The framework includes EU-wide targets and policy objectives for the period from 2021 to 2030 to enable the EU to implement its commitments under the Paris Agreement. One of the key targets is to cut greenhouse gas emissions by at least 40% from 1990 level.

Air pollution is shown to be the most important reason for serious illnesses causing early deaths, such as heart and lung diseases, stroke, lung cancer. Around 400.000 premature deaths per year in the EEA39 (excluding Turkey) are caused by those diseases [4]. Microscopic pollutants damage the lungs, heart, and brain by penetrating deep into the respiratory and circulatory system. According to World Health

Organization, nine out of ten people breathe polluted air that is the reason for 7 million deaths every year because half of the world has no access to clean fuels and technologies [5]. Not only people who have a chronic disease but also healthy people may have problems because of air pollution. For example, breathing difficulties or respiratory irritation may be caused as they are exposed to air pollution in the long-term. Respiratory system diseases such as decreased lung function and asthma occur in those exposed to either short or long term air pollution.

Air pollution can affect directly natural ecosystems and biodiversity. Nitrogen oxides (NO , NO_2) cause degradation by feeding excessive amounts of nitrogen nutrients into terrestrial and aquatic ecosystems. As a result, new species invade the region and the diversity of species changes. The high presence of NO_X and SO_2 causes acidification of soil and water sources, leading to loss of biodiversity. As for ground-level O_3 , it lowers the growth rates of crops and trees.

Particulate matter (PM) is made up of small particles in the air, such as dust, soot, and drops of liquids. In urban areas, the majority of PM is created directly from the burning of fossil fuels by automobiles, power plants, non-road type of equipment, and industrial facilities. There is a relationship between increasing adverse health effects and particulate matter even at lower-level concentration. Particles less than 10 microns in diameter are called coarse particulate matter (PM_{10}) and they cause nasal and upper respiratory tract health problems. They are one of the main reasons for premature death from heart ailments, lung disease, and cancer.

Nitrogen Oxides, like Nitrogen Oxide and Nitrogen Dioxide, are produced by the transportation sector. Since NO is converted to NO_2 in sunlight, NO_X which is a combination of NO and NO_2 can be measured as a parameter of the air pollution. NO_X exists in high concentrations around roadways and it can lead to asthma and bronchitis and can cause a high risk of heart disease.

Although ozone in the high atmosphere protects us from ultraviolet radiation, ozone at ground level irritates the respiratory system by being a part of smog. It is

generated by the reactions of volatile organic compounds and nitrogen oxides, caused by the combustion of fossil fuels. Exposure to ozone causes coughing, chest pain and throat irritation in short-term exposure, as well as chronic obstructive pulmonary disease, and decreased lung function in long-term exposure.

Sulfur Dioxide (SO_2) spreads out into the air from vehicles (ship engines, heavy equipment, diesel equipment) that burn fossil fuels containing sulfur (like coal). Eye irritation, asthma, respiratory infections, and cardiovascular system failures can be caused by Sulfur Dioxide. Besides, the combination of SO_2 and water becomes sulfuric acid which is the major component of acid rain and it leads to deforestation.

In light of this information, air quality standards are set as an important component of national risk management and environmental policies in order to protect the public health of citizens. Predicting the air pollution allows to detect the abnormality in the region in advance so that some health improvements can be performed by different air pollution reduction scenarios, necessary precautions can be taken, warnings and health practices can be taken for the residents.

A smart city is an urban place that deploys different types of electronic devices and various sensors to collect data from the environment and citizens. The information derived from that data are used to manage resources and services efficiently in traffic, transportation systems, information systems, community services, and utilities [6, 7]. It is a concept to develop a city using information and communication technologies to enhance the quality and performance of urban services in terms of economy, environment, people, mobility, governance, etc. The objectives of smart cities are to (1) improve the quality of city life, (2) promote sustainable development, and (3) raise public welfare [7, 8]. For more than a decade, since recent social and environmental developments (climate change, economic restructuring, urban population growth) have increased interest in smart city studies [9], city systems have been developed to digitize city information. This information is processed and made meaningful in software platforms in order to improve urban services such as trans-

portation, health, education, culture, environment, etc. and to work in coalition in terms of such services. Furthermore, enormous amount of information is gathered from various disciplines by sensors located at many points of the city.

The European Union has developed strategies and programs to manage smart urban growth in order to deal with the high energy consumption and greenhouse gas caused by the growing population in the metropolitan areas and to provide sustainable prosperity. European Innovation Partnership on Smart Cities and Communities (EIP-SCC) is an initiative that is supported by the European Commission; brings together cities, industries, small businesses, banks, and research institutions. It is aimed to improve urban life by bringing sustainable solutions to city problems in different fields such as energy, transportation, and communication [10]. Smart city technologies offer authorities the opportunity to monitor city residents, infrastructures and interact with them directly. With these technologies, it is aimed to increase the service quality and performance, as well as to reduce costs and resource consumption by real-time responses.

The process of smart city data analysis can be approached in four stages: data acquisition, data preprocessing, data analysis, service provision [11, 12]. (1) Data acquisition is the process to collect and store smart city data coming from various domains and sources; (2) data preprocessing is use of techniques to transform raw data into an understandable format; (3) data analytics is to perform intelligent analysis by different machine learning techniques to discover knowledge about domains and sources for applications; (4) service provision is the development of intelligent services and applications.

Internet of Things (IoT) is a concept of a communication network that connects any device (sensors, microcontrollers, transceivers, etc.) which collects and shares data about the environment around them, to the Internet and to other devices. The foresight that the number of internet-connected devices will exceed 50 billion in the coming years [13] has made the IoT paradigm a global infrastructure for the

information society. The aim of IoT concept is to make the Internet more immersive and pervasive by easy access and interaction with various everyday life devices and sensors used in smart city systems [14], for environment monitoring, smart health care, public safety, intelligent transportation, etc. [15, 16, 17, 18, 19, 20].

United Nations reported that more than half of the world's population has been living in urban areas since 2018 and the number of people living in urban areas will make up two-thirds of the total world population in 2050 [21]. In this case, people living in urban areas become customers who need better conditions in terms of environment, transportation, health, and so on. In recent years, the rapid development of IoT makes accurate environmental monitoring possible with updates on air pollution in real-time. IoT devices generate a huge amount of data continuously from various places in the world. Artificial intelligence-based air pollution prediction models for smart cities are able to process data that comes from a number of IoT sensors placed over a large area. Such models use machine learning algorithms in order to learn the correlation between features and make predictions accurately.

1.3 Subject and Scope

In recent years, many countries have taken important steps towards developing smart cities such as Santander, Barcelona, Singapore [22, 23, 24]. Many authorities began to share data on the open data platform as a part of transparency initiatives, which can be collected and documented to monitor the city. Numerous IoT devices used in the smart city structure constitute the data sources. Artificial intelligence techniques are used predominantly in computer science to process enormous amounts of data, called Big Data, and to obtain useful information in order to develop efficient applications [25]. Large scale sensor networks included in IoT and smart city concepts have led to the emergence of Big Data. The integration and use of Big Data from the large sensor network in smart city applications is a major research challenge [26].

Nowadays, air quality monitoring and forecasting projects have been carried out [27] and such works point out some requirements for smart city air pollution applications. Changes in city conditions and behavior of people living there over time may adversely affect the performance of the model, which is trained with old data. It is necessary to train and optimize the model periodically with current information [28].

Recent studies [29, 30, 31] show that the features of particulate diffusion related to time and space must be considered to accurately model air pollution prediction. Particulates in the air can pollute the surrounding environment by standstill at a fixed location or moving between gaps in the air [29]. The particulate concentrations are the main temporal factors that reflect the correlations between regions [30]. The analysis only in the time domain may cause disregard of impacts between regions. On the other hand, the analysis only based on spatial dependency may cause ignorance of the effects related to the diffusion of particulates over time. Therefore, to design a reliable model, consideration of the spatial features and temporal features together as proposed in this thesis, is needed in order to detect the regional historical trend of air pollution and the interaction of pollutants between regions.

Data fusion is a process of getting data from multiple sources and combining it in a way to achieve accurate and useful representation of real world environments. The propagation of air pollutants is affected by meteorological conditions (like temperature, humidity, wind direction, and strength) [32, 33] and temporal features are determined by concentrations of pollution particulates [34, 35]. Such information is gathered from various sensors that are deployed into the related city and must be used from the prediction model together and at the same time.

Artificial intelligence and machine learning-based techniques are used for modeling, learning, search, and optimization solutions in smart city applications [36, 11, 37, 29, 38, 39]. Machine learning algorithms are computational methods which improve the learning performance of machine from complex data in order to be capable of

solving nonlinear problems. The accumulation and easy access to big data in many areas and the high accuracy of machine learning methods on these data have led to a tendency to various machine learning methods in air quality studies [37, 29, 38, 39, 20]. Deep learning is a neural network-based machine learning method that consists of multiple processing layers to learn representations of data with multiple levels of abstraction and is able to discover complex structures from data in raw forms without a sophisticated feature engineering. As a result of its deep modeling feature, it can be used as an effective approach in solution of nonlinear problems.

The types of air pollutant sources have increased with urbanization, the dynamics of the concentration of pollutant particles in the air has become more complex [37, 40]. Although numerical analysis and machine learning methods are widely used in air pollution prediction [27, 31, 41] there are several drawbacks that affect negatively prediction capability. Atmospheric condition is too complex to be represented by certain regular behavior. Also, traditional machine learning methods ignore deep relations with spatial information or temporal changes. The data used for air pollution prediction is generally nonlinear, so a nonlinear modeling approach is more suitable for such data.

Deep learning is a machine learning technique that can perform classification and regression tasks directly from data by using large data sets. Deep neural network architectures may contain a large number of hidden layers. Deep learning methods use mostly neural networks and one of the most popular deep learning network is Deep Convolutional Neural Networks (D-CNN). Feature representation of input data are obtained through hidden layers by increasing the complexity of the learned features. At the last layer, all features are combined to obtain feature maps that represent the input.

Most machine learning methods require manual feature extraction. The extracted features are used to create a model for classification or regression. Deep learning extracts relevant features automatically from input data. Also, deep learning models

can be called “end-to-end learning” where raw data are given to the network and the network model performs a task (classification or regression), thus, the learning process happens automatically.

Convolutional Neural Network (CNN) is a special type of neural networks which is inspired by primate’s visual cortex structure. The topology of CNN is composed of multiple learning stages that include a convolutional layer, nonlinear processing units. Each layer performs multiple transformations by convolutional filters. Non-linear processing units uses the output of kernel filters to provide learning from abstraction and embed nonlinearity in the feature space. Different patterns and activations are generated by nonlinearity for different responses to make possible learning of semantic differences in data.

The topology of Convolutional Neural Network (CNN) is composed of multiple learning stages that include a convolutional layer and nonlinear processing units where each layer performs multiple transformations by convolutional filters. Nonlinear processing units use the output of kernel filters to provide learning from abstraction and nonlinearity in the feature space. Different patterns and activations are generated by nonlinear processing for different responses to make possible learning of semantic differences in data. CNN is capable of extracting low-level features, mid-level features, and high-level features automatically. Therefore the use of CNN reduces the need for a synthesis of different feature extractors.

In feedforward neural network models, the input is passed through the network to give the output. Input is labeled for supervised learning. In such networks, the training progresses by updating the weights until the error in the output decreases to a sufficiently small value.

Recurrent Neural Networks (RNN) are a type of neural network capable of holding the state information, as well as producing an output according to an input. This capability makes the RNN structure applicable to data that has a certain order. In feedforward networks all outputs are independent of each other, on the other hand,

the output depends on the previous computations in RNN.

RNN evaluates the inputs on the previous inputs and generates output by combining the current and the previous information. Each RNN neuron has a feedback loop to feed the output of the previous time step as the input of the next state. Recurrent networks are used to understand the sequential information of input data such as various time dependent sensors data. RNN performs the same task for each element of input sequence by a memory that captures information to produce the correct output.

Although the theory points out that RNN is capable to use information in long sequences, it is limited to looking back only a few steps to connect previous information to the present task in the practice. If the gap between the relevant information and the needed place is large, RNN becomes useless to learn the connection between the information in different steps. This problem is called long term dependencies.

The Long Short-Term Memory (LSTM) is capable of reading, writing, and deleting information from its memory, like the memory of a computer, thus it enables RNN to remember their inputs over a long period. LSTM uses a different function from RNN to compute the hidden state by combining the previous state, the current memory, and the input. LSTMs have memory cells and each cell works with gates that decide whether to store or delete information to learn whether the information is important or not over time. As for another advantage of CNN and LSTM, those models work based on parameter sharing. The main purpose of parameter sharing is to reduce the parameters that the model has to learn. In CNN, convolutional filters share the parameters while it is done by sharing the same weights by RNN in sequence learning.

Many machine learning methods work well only when the training data and test data have the same feature space and the same distribution. Most statistical and machine learning models require the collection of new training data when the distribution changes. This can cause a more difficult and expensive process to recollect

training data and efforts to rebuild the learning model in real world applications. It is assumed that training and test data which are used in machine learning techniques are obtained from the same domain and have the same distribution. This assumption does not apply to large systems with resources distributed over various spatial locations.

Transfer learning focuses on methods that extract information from a data set and reapply it in another data set. The data in different regions are represented by different feature distributions and transfer learning provides that the model is able to transfer knowledge from a source domain to a target domain where the domains have a different distribution. In this study, transfer learning is used to support the proposed CNN+LSTM deep neural network. The main purpose is to transfer weights, so the features, from the source domain to the target domain in order to improve the prediction accuracy in different cities.

1.4 Contribution and Impact

Smart city applications are supposed to process spatial and temporal data [28, 29, 42, 43] as particulates can pollute the surrounding environment by standstill at a fixed location or moving between gaps in the air over time [29]. Air pollution depends on meteorological conditions such as emissions, wind direction, wind strength, etc. [29, 32, 33]. The analysis only in the time domain may cause disregard of impacts between regions. On the other hand, the analysis based on only spatial dependency may disregard diffusion of particulates over time. Additionally, changes in city conditions and behaviour of people living there over time may affect negatively the performance of the model that is trained with old data. Because each city's data have a different marginal probability distribution, it is necessary to train and optimize the model at regular intervals with current data [28]. Another challenge about air pollution prediction is that the system should alert residents of the city in a short time, send necessary warnings to relevant authorities in order to take precautions and interfere with the pollution. It is aimed to solve these challenges

by developing a CNN+LSTM deep neural network model for hourly air pollution prediction. The historical data from various locations in a city is processed and the concentration of particles in the same locations is output as the next hour. In addition, the proposed model is supported by the transfer learning method to obtain a deep learning structure with the optimal set of hyperparameter values to use in various cities.

In this study, data used by the model are collected from different cities in the world and various points of these cities. Data includes information about both pollutant concentrations and meteorological variables. One advantage of artificial intelligence is the capability to extract relevant complex features from input data automatically and learn the correlation between them for a successful prediction. CNN improves the generalization and learning ability of the model by reducing the size and complexity of data. Studies in the literature often include feature engineering to determine the features to be input into the model. In this study, the deep learning model automatically learns the features and the correlation between them to extract the structure of the inputs.

Another advantage is that AI is capable to run different types of machine learning algorithms at the same time to improve the learning performance of the air pollution prediction system. Such AI models can process both spatial and temporal features simultaneously from big data in a short time. The input of the proposed system consists of both spatial and temporal information of the pollutants in the city. While spatial features are based on the locations of the sensors, temporal features are based on the historical hourly concentration of the pollutants. CNN is used to extract spatial features from both pollutant concentrations data and meteorological data with separate sensor locations. Various sensor data collected from different points of a city are processed in order to predict not only local pollution that affects a small area but also propagation pollution that affects the wider precinct. Information from sensors located in different locations of a city is processed at the same time, thereby revealing the air pollution relationship between locations. Meanwhile, LSTM is used

to extract temporal information characteristics from different time series data and it consolidates the nonlinear relationship between multivariable time series and air pollutants. At the end of the study, not only the temporal effects on the concentration of a pollutant in the next hour but also spatial interaction contributes to the prediction performance.

Spatial features of target locations are used to increase the prediction sensitivity and explicitly consider meteorological impacts for pollutant propagation. Thus, the proposed model uses temporal information based on historical data of target locations with high spatial or temporal similarity. There are two input structures for the use in the system; 2-dimensional and 3-dimensional array. While 2D input contains the historical information about only target pollutants at sensor locations, 3D input contains both the historical information about the target pollutant and other pollutants together. Thus, the 3D input structure ensures the model to consider the effect of other pollutants (and meteorological data) on the presence of a pollutant and to give prediction values for more than one pollutant as output at the same time.

Transfer learning enables the model, which successfully performs a specific task on a data (source domain), to be used on another data (target domain) with a different distribution from the previous data. The task on the target domain can be the same as the previous task or a new task related to the previous one. For transfer learning, the last one or two learning layers of the pre-trained model are usually changed and the model is tested on the target data after a short training process with the training set of the new data. In this study, a pre-trained network that successfully predicts the pollutant concentration in a certain city is supported by transfer learning to be used to predict the same or different pollutants' concentration in another city. The pre-trained network is tested directly on the target city, then a short training process is applied to the network and the performance on the test data are measured. In this way, the contribution of the short training with the training set created from the target domain to the prediction accuracy is investigated.

2. LITERATURE

Summary of machine learning-based air quality studies in the literature is given in Table 2.1. In these studies, the effects of spatial and temporal features on prediction performance are analyzed and several machine learning algorithms are compared.

[34] compares three machine learning algorithms for accurate air pollution prediction. Support Vector Machine (SVM), Artificial Neural Network (ANN), and model trees (M5P) use historical pollutant concentration (O_3 , NO_2 , SO_2 , H_2O) and meteorological data (temperature, humidity, wind speed) from June to August 2013. The Air Quality Monitoring (AQM) network, a pilot initiative of Qatar Mobility Innovations Center (QMIC), is used for data collection. Algorithms are run for univariate and multivariate data. Prediction results with multivariate data are better than univariate data because the dependency between target gases and other features can be included in the prediction processes. Using M5P and SVM reduces RMSE by 35.4%, 30.5%, and 58.7% compared to ANN in multivariate data for ground level ozone (O_3), Nitrogen Dioxide (NO_2) and Sulfur Dioxide (SO_2) respectively. While the best PTA values in order of Ground-Level Ozone (O_3).

In [38], the prediction model consists of 1D-CNN (1-Dimensional CNN) and Bi-GRU (Bi-directional GRU). they use the UCI machine learning repository database for $PM_{2.5}$ data of the US Embassy in Beijing and Capital International Airport meteorological data. Even though the embassy and airport are 17 km apart, the same weather is experienced. Pollution, dew point, wind direction, wind speed and temperature, air pressure, snowfall, and rainfall are selected as features from hourly data between the 1st of January, 2010 to 31st of December, 2014. The model is trained and tested with a combination of different features and the model gains bet-

ter performance with inputs of pollution, temperature, wind speed, wind direction, dew point. When the other features are included in the input, the complexity and difficulty of learning increased. It is noted that the variants of RNN (LSTM and GRU) have better performance than RNN, and they are better to learn local trend information and long term dependencies while CNN is better in local feature learning and dimensionality reduction. Also, using Convolutional Based Gated Recurrent Unit (CBGRU) reduces RMSE and MAE by 47.65% and 37.32% compared to Support Vector Regression (SVR), while the SMAPE values for CBGRU and SVR are 0.26 and 0.21 respectively.

[29] proposes a spatial-temporal deep neural network (ST-DNN) model to predict air quality. The model uses historical data including temporal information, the concentration of pollutants ($PM_{2.5}$ and PM_{10}), and meteorological characteristics (wind speed, wind direction, temperature, humidity) as input, then they predicted the future $PM_{2.5}$ concentration as output. Two real-world data sets (76 locations in 23 cities in Taiwan and locations from Beijing) are used in experiments. Data include measurements from January 2014 to September 2017 and it is divided by 2:1 ratio as the training set and test set. The model is constructed by combining a Convolutional Neural Network (CNN), Adaptive Temporal Extractor (ASE), and Long Short-Term Memory (LSTM). Air quality and meteorological condition data are input to ASE and LSTM whereas terrain information is input to CNN. Since the convolution layer can extract the temporal delay factor from surrounding features by learning spatial information, the proposed model provides a long time frame prediction.

In [44], they apply a regression technique by ANFIS modeling to conduct a time series prediction for air pollution. According to the model, the concentration of an air pollutant in the current time depends on the concentration of pollutants in the past 5 days. O_3 , SO_2 , CO , NO_2 are considered with 24 hours time series length. The input data are separated randomly into three data sets for training, validation, and test. While the training data set contains 305 points, the validation test sets

contain 25 points. They use two approaches, regression modeling, and ANFIS. The ANFIS model is constructed with an input layer, one hidden layer that consists of fifteen membership functions, and an output layer. The input layer is with five input variables that represent the contamination rate for consecutive five days. The output layer gives a single output variable that is the predicted contamination rate for the next day. The RMSE of ANFIS was measured as 0.17 for CO , 0.25 for SO_2 , 0.20 for O_3 , and 0.16 for NO_2 . The RMSE of the semi-experimental regression model was measured as 0.21 for CO , 0.27 for SO_2 , 0.24 for O_3 , and 0.16 for NO_2 . Therefore, the ANFIS model is observed as a more accurate prediction model for air pollution than the regression model. The comparison between error values related to different pollutants points out that the ANFIS model designed for NO_2 has the least error value for short-term contamination prediction.

[45] uses Hidden Markov Models (HMM) and LSTM DNN prediction model with airborne-pollution data sources in Melbourne Urban Area (Victoria, Australia). Air pollutants ($PM_{2.5}$, PM_{10} , SO_2 , CO , and O_3), Air Quality Index (AQI), and meteorological attributes (temperature, relative humidity, wind speed, wind direction) are used as part of the model. The data collected from January 2017 to January 2019 are normalized to values between 0 and 1 and input to the model, the outcome is the predicted value set of the target pollutant for different time lags. The RMSE values are measured as 2.62 in Traralgon, 6.77 in Mooroolbark, 1.92 in Alphington, and 3.85 in Melbourne. When the prediction system was applied in the Melbourne Urban Area (Australia), the prediction accuracy increased up to 3% for some of the monitoring stations.

Also in [37], they combine CNN and LSTM as the air quality prediction model. CNN convolutional layer is used to extract features and reduce the complexity by its sharable local weights while LSTM is used for time series pollutant prediction with long-term dependency. They select Shanghai as the target city and collect data manually between 2015 and 2017 with pollutant and meteorological information. The data of 2015-2016 is used as training data while the data of 2017 is used as

test data. The input features are $PM_{2.5}$ concentration, temperature, wind speed, wind direction, humidity, and other pollutant concentration. The output is the predicted sequence of $PM_{2.5}$ values for the target city Shanghai. With the comparison of the CNN+LSTM model with RNN, CNN, and LSTM, the performance of the CNN+LSTM model is the best for long-term sequence prediction. RNN and CNN show the same value for RMSE where CNN+LSTM (proposed model) reduces RMSE by 41.45% compared to CNN and RNN.

[39] uses a model that combined bi-directional LSTM (BLSTM) and Inverse Distance Weighting (IDW) to predict the concentration of $PM_{2.5}$ in Guangdong, China. Hourly data are collected from 100 locations in Guangdong to build time series samples between January 1 and December 31, 2017. Unlike CNN-LSTM combined models that are temporal first, IDW-BLSTM is a spatial first mechanism. Results show that the proposed model IDW-BLSTM, BLSTM, CNN-LSTM, and LSTM reach RMSE values close to each other.

In [46], they compare Multilayer Perceptron (MLP) and CNN for PM_{10} concentration prediction with different model hyperparameters. The data set from a public database of a Mexican government (Secretariat of Environment (SEDEMA by its acronym in Spanish)) contains the hourly measurements of 7 different variables in Mexico City. The information about PM_{10} concentration and meteorology (temperature, wind speed, wind direction, relative humidity, solar ultraviolet type A, solar ultraviolet type B from 2000 to 2018 is used for the prediction system. They observed 13 different models built based on MLP and CNN with various hyperparameter combinations and various activation functions (sigmoid, ReLU, linear). The best prediction result in terms of test RMSE is obtained with 2-Dimensional CNN (2DCNN) with sigmoid activation function. According to the comparison of the RMSE values observed with MLP and CNN, The lowest RMSE value was measured as 18.77 with the 2DCNN structure.

[47] shows the performance comparison between univariate and multivariate pre-

Table 2.1: Proposed architectures and their performances in literature

Reference	Features	Output	Architecture	Performance
Shaban <i>et al.</i> [34]	Concentration of target pollutant (O_3 , NO_2 , SO_2)	Concentration of target pollutant	ANN	RMSE(O_3)=26.4
				RMSE(NO_2)=25.7
				RMSE(SO_2)=74.1
	Concentration of target pollutant and meteorological data	Concentration of target pollutant	ANN	RMSE(O_3)=11.6
				RMSE(NO_2)=25.7
				RMSE(SO_2)=107.6
Tao <i>et al.</i> [38]	Concentration of $PM_{2.5}$ and meteorological data	Concentration of $PM_{2.5}$	RNN	RMSE=20.94
			LSTM	RMSE=17.31
Soh <i>et al.</i> [29]	Concentration of pollutants and meteorological data	Concentration of $PM_{2.5}$	ST_DNN with ANN, LSTM and CNN	MAE=3.40
Zeinalnezhad <i>et al.</i> [44]	contamination rate of target pollutant for consecutive five days	the predicted contamination of target pollutant rate for the next day	ANFIS	RMSE(SO_2)=0.25
				RMSE(O_3)=0.20
				RMSE(NO_2)=0.16
			Semi-experimental Regression	RMSE(SO_2)=0.27
				RMSE(O_3)=0.24
Qin <i>et al.</i> [37]	Concentration of air pollutants and meteorological data	Concentration of $PM_{2.5}$	CNN	RMSE=30.66
			LSTM	RMSE=17.95
			CNN+LSTM	RMSE=14.30
Ma <i>et al.</i> [39]	Concentration of $PM_{2.5}$	Concentration of $PM_{2.5}$	ANN	RMSE=11.20
			LSTM	RMSE=8.98
			CNN+LSTM	RMSE=8.40

diction for PM_{10} concentration with LSTM. The PM_{10} concentration data set is provided by the Regional Agency for Environmental Protection of Abruzzo (ARTA) and contains time series measurements from January 2009 to December 2017 in Pescara, Italy. The study is done for both univariate and multivariate prediction by using the dependencies of PM_{10} on the various combinations of local environment features (temperature, humidity, pressure, tropopause height, cloud fraction, 24 hours accumulated precipitation). The model reached the lowest RMSE as 9.36 with multivariate prediction and the LSTM is found as an efficient time series prediction model. Additionally, an increase in the number of features causes a decrease in prediction performance as it increases the complexity of the input.

As for transfer learning methods for air quality, [48] proposes a model called Multi-view Transfer Learning Semi-Supervised learning for Air quality Estimation (MT-

SAE) to transfer features from urban areas to non-urban areas. They use three types of features that are terrain, spatial, and temporal. Real air quality data set (Air Quality Index (AQI), PM_{10} , $PM_{2.5}$, SO_2 , NO_2 , CO, O_3) and meteorological data set (weather, temperature, wind speed, humidity) are collected hourly from three cities in China (Hangzhou, Ningbo, Wuxi). Also, terrain data set is taken from OpenStreetMap with geographical shapes. Results show that adding terrain features in the learning process doesn't make a significant improvement in prediction for urban areas.

In [49], they implement a model called Flexible multimodal tRANSfer Learning (FLORAL) that is based on multi-modal transfer learning in order to transfer knowledge between cities (from Beijing to Shanghai). The data set contains four main subjects; road networks (endpoints, length, level of capacity), Point of Interests (POI) (name, address, coordinates, category of a venue), hourly meteorological data (weather, temperature, humidity, barometer pressure) and taxi trajectories from February 2nd to May 26th, 2014. The model classifies output as six categories according to the air quality level. As a result, FLORAL outperforms baselines (some of which use transfer methods) up to 50%.

Lastly, [50] uses Bidirectional Long Short-Term Memory and transfer learning to predict air quality and compare its performance with other commonly seen models. They conducted a case study in Guangdong, China. The data are collected from all the monitoring stations in the city for three years and contain the hourly $PM_{2.5}$ concentration. Transfer learning helps to improve the prediction accuracy of BLSTM at larger temporal resolutions up to 40%.

3. METHODOLOGY

The first aim of the thesis is to predict the concentration of pollutants in the next hour in more than one location in a city at the same time. The second aim is to transfer knowledge between data sets that have different distribution for air pollution prediction accuracy in cities. The study can be split into three parts according to the type of input-output representation:

Method-1: To predict the concentration of one pollutant, the 2-dimensional array containing the information of one pollutant in past hours is input to the model. Input consists of the historical data of one pollutant in all locations; output is the predicted concentration of that pollutant in all locations in the next hour.

Method-2: To predict the concentration of one pollutant, the 3-dimensional array containing the information of all pollutants in past hours is input to the model. Input is the historical data of all pollutants in all locations; output is the predicted concentration of one pollutant in all locations in the next hour.

Method-3: To predict the concentration of all pollutants at the same time, the 3-dimensional array containing the information of all pollutants is input to the model. Input is the historical data of all pollutants in all locations; output is the predicted concentration of all pollutants in all locations in the next hour.

Time series data are provided from open data services that are offered by the Municipality of Barcelona and the Republic of Turkey Ministry of Environment and Urbanization. İstanbul, Kocaeli, and Barcelona are chosen in order to evaluate the performance of the proposed methods in different cities. Although information about air pollution is available for all cities, meteorological data are available only

Table 3.1: Explanation of the methods according to the input-output type

Method	Input Structure	Input	Output
Method-1	2-dimensional array	Historical information of target pollutant (univariate input)	Predicted value for the concentration of target pollutant (univariate output)
Method-2	3-dimensional array	Historical information of all pollutants and meteorological data (multivariate input)	Predicted value for the concentration of target pollutant (univariate output)
Method-3	3-dimensional array	Historical information of all pollutants and meteorological data (multivariate input)	Predicted values for the concentration of all pollutants (multivariate output)

for İstanbul and Kocaeli. O_3 , NO_2 and PM_{10} were selected as target pollutants for the city of Barcelona, while PM_{10} , SO_2 and NO_X were selected in the city of Kocaeli. Since the information of all these pollutants is available in İstanbul data, two separate time series databases have been created for İstanbul: (1) data set containing features of the same pollutants as Barcelona and (2) data set containing features of the same pollutants as Kocaeli.

In both air quality and meteorology data, there are “duplicate data” at the beginning, which means that some consecutive hours have the same values. Firstly, all duplicate data are cleaned and linear interpolation is applied to fill in missing values in the data, then the data are normalized to scale the values into a certain range. Data normalization is a method of preprocessing that scales the data to a certain range. Data normalization has an important place in machine learning methods since it reduces the data redundancy caused by the wide variation of the range of the values in the data [51]. Numerical data obtained at different scales are scaled within a specific range to make the distribution of the data regular. Thus, the bias effect of the features with relatively larger magnitudes in the data set is removed.

Without normalization, features with larger quantities would be more dominant on the analysis results since the clustering depends on a distance measure.

There are two types of input for the neural network: 2D and 3D array. These input structures hold both spatial and temporal features together. Where the 2D array consists of only the historical information of target pollutants, the 3D array contains all pollutants' concentration and meteorological information. Thus, the model uses both the interaction between different pollutants and the effects of meteorological features (temperature, pressure, wind speed, and direction, etc.) on particles in a location and neighboring locations. Table 3.2 explains the data properties used for the methods in this thesis.

Table 3.2: Explanation of the data properties used for the methods

Properties		Method-1	Method-2	Method-3
Input Structure	2D Input	YES	NO	NO
	3D Input	NO	YES	YES
Data Normalization	Normalized Data	YES	NO	NO
	Non-Normalized Data	YES	YES	YES
Meteorological Data		NO	YES	YES
Transfer Learning		NO	YES	YES

The first step of machine learning model design is to determine the layers and their properties. Therefore, the number of layers, filter type and the number of filters, and the number of hidden units are determined for the CNN+LSTM deep learning model. Then, various pooling methods are added to the CNN part of the model (max pooling, average pooling, etc.) and optimal architecture and hyperparameter values are determined by train-testing with different activation functions (ReLU, sigmoid). For transfer learning, weight transfer from the source domain to the target domain is made by changing the fully connected layer which is the last learning layer of deep neural network.

3.1 Data

Barcelona Air Quality Data: Air quality historical data for Barcelona is taken from The Open Data BCN which is a service managed by the Municipal Data Office [52]. Real-time hourly measurement is made for O_3 (tropospheric ozone), NO_2 (Nitrogen dioxide), PM_{10} (Suspended particles) by all the stations throughout Catalonia. Note that the historical resources include only data from the stations in Barcelona (Ciudadella, Eixample, Gracia, Palau Reial, Poblenou, Sants, Vall Hebron). Data consist of hourly information from 06/13/2018 to 01/31/2019 and it is collected from 3 sensors located in 7 different locations of Barcelona. This information includes the location label of sensors, date and time, and the concentration of air pollutants (O_3 , NO_2 , PM_{10}). The distance between selected locations in Barcelona is not more than 5 km.

Kocaeli and İstanbul Air Quality Data: Hourly historical data provided by Republic of Turkey Ministry of Environment and Urbanization [53] contains the concentration of various pollutants (Sulfur Dioxide(SO_2), Nitrogen Oxides(NO_X), Tropospheric Ozone (O_3), Particulate Matters (PM), etc.) and meteorological data (temperature, relative humidity, air pressure, wind speed and direction) for cities Kocaeli and İstanbul. The data are collected from sensors distributed in different locations in each city. Kocaeli data are for the dates between 11/14/2017 17.00 and 4/11/2020 23.00 (total of 21103 hours) and İstanbul data are for the dates between 01/01/2015 00.00 - 04/11/2020 23.00 (total of 46272 hours). Kocaeli data consist of information collected from 7 locations in the city (Alikahya, Dilovası, Gebze, Gölcük, Körfez, Yalova, Yeniköy) and the data from 3 locations (Alikahya, Gebze, Körfez) are chosen for this study. İstanbul data consist of information collected from 7 locations in the city (Başakşehir, Esenyurt, Kağıthane, Silivri, Sultanbeyli, Sultangazi, Şile) and the data from 3 locations (Silivri, Esenyurt, Sultangazi) are chosen for this study. While selecting the locations in Kocaeli and İstanbul, access to the information of target pollutants and missing data percentage were taken into consideration. The approximate distance between selected locations in Kocaeli and

İstanbul is as follows. In Kocaeli, the distance between Gebze and Körfez is 25 km, the distance between Gebze and Alikahya is 18 km. As for İstanbul, the distance between Silivri and Esenyurt is 37 km, the distance between Esenyurt and Sultangazi is 17 km.

3.2 Time Series Data Set Preparation

3.2.1 Rolling-Window

For the time series problem, the “rolling-window” method [54] is used in order to create time series input for the deep learning model. This method constructs one sample for each time record t by using the values $[t_0 - d, t_0)$ as the features of the target value at t_0 . The example in Figure 3.1 shows how the rolling window method works. If d and s represent the frame size and the step size respectively, the values of the past d time record points are taken as features and the value of the time step t_0 is taken as the target value, then the frame is slid s steps for the next sample.

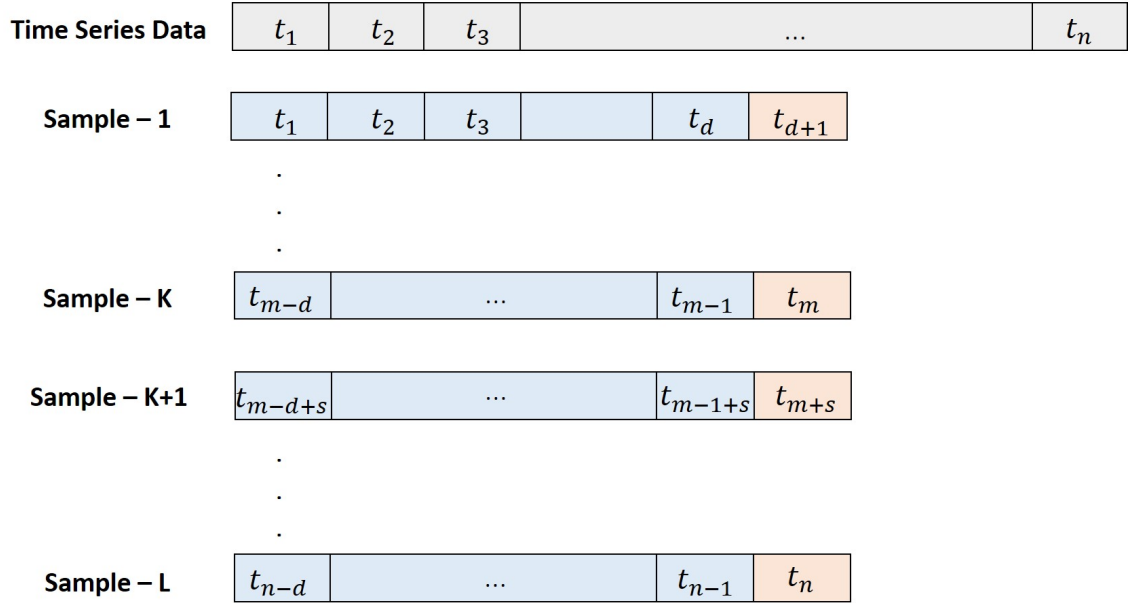


Figure 3.1: Rolling window method for the time series data

3.2.2 Input Structure of CNN

After the rolling-window method, a 2D sample model (given in Figure 3.2) is constructed in order to be used as input of the Convolutional Neural Network.

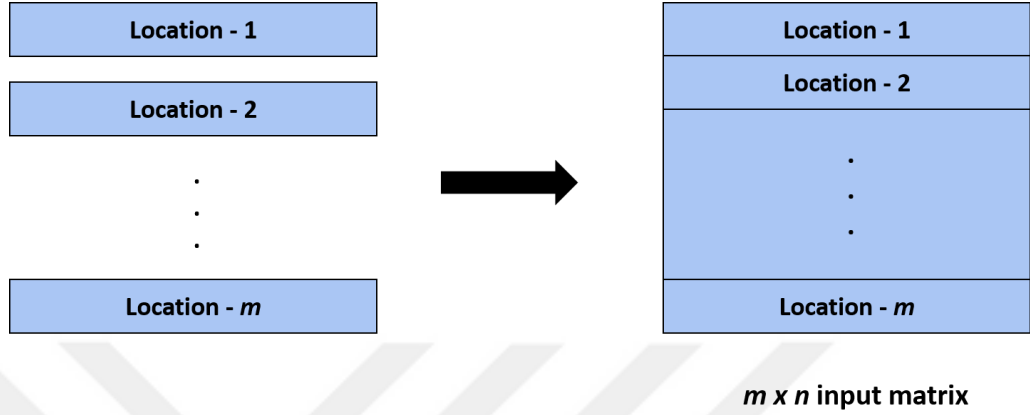


Figure 3.2: $m \times n$ input matrix for Convolutional Neural Network where m is the number of locations and n is the number of time step.

For the prediction with Method-1, the matrix in Figure 3.2 is input to the model to obtain an output with $m \times 1$ size that represents the predicted values of concentration of the pollutant in each location. Because of CNN's ability to process 3D structures, the input shape can be rearranged as given in Figure 3.3 where m , n , and k represent the number of location, the number of the time step, and the number of pollutants respectively. The 3D input structure is used for the prediction with Method-2 and Method-3.

3.3 Artificial Neural Networks

Artificial Neural Networks are computational models that simulate the way biological neural systems perform and composed by a set of artificial neurons connected to each other. Each connection between neurons has a weight that is the strength of the connection. Weights representing the synapses in biological neural networks decide how much influence the input has on the output. The goal of the training process is to update those weights to decrease the loss (error) at the output. Bias (or offset) is an extra input to neurons and used with the purpose that there is an

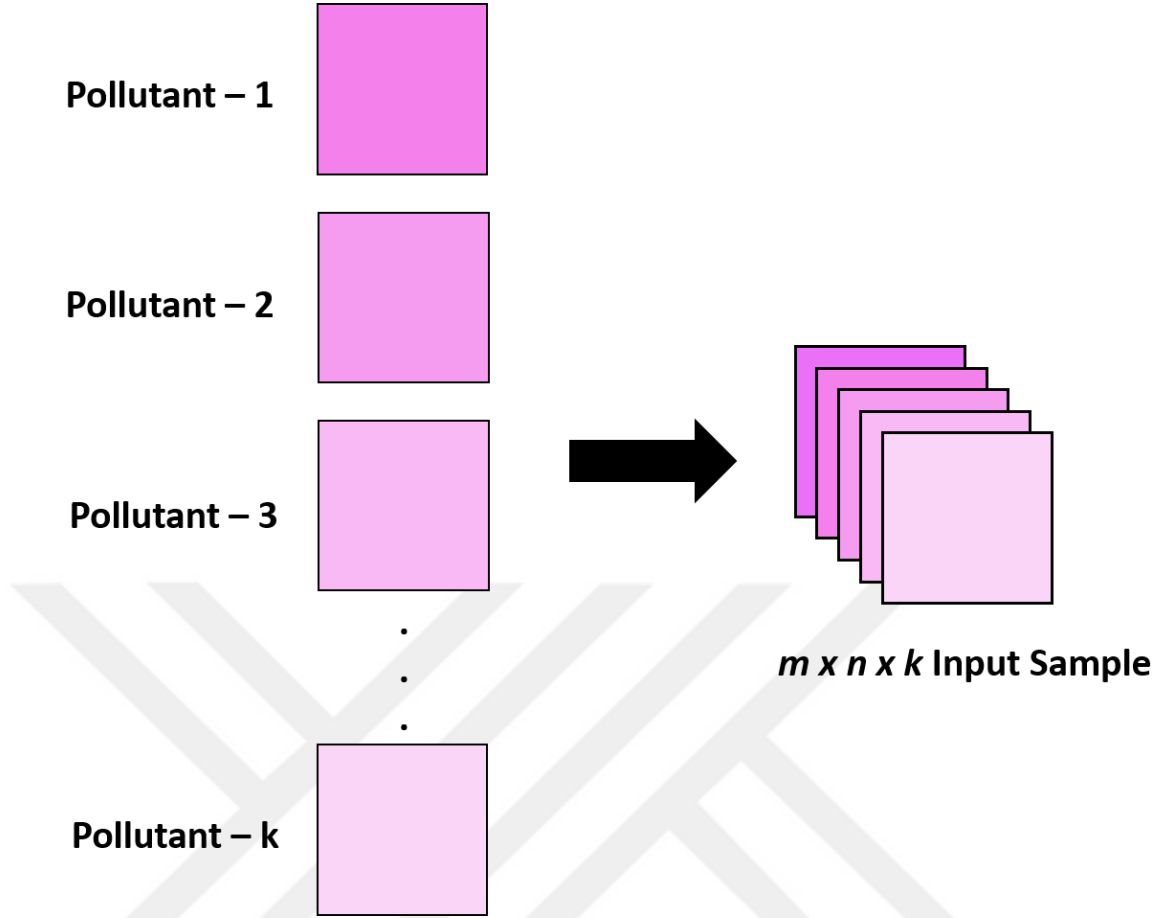


Figure 3.3: $m \times n \times k$ input matrix for Convolutional Neural Network

activation in the neuron even when all the inputs are zero. The activation function (or transfer function) is used to introduce nonlinearity to neural networks. In literature, there are several activation functions such as hyperbolic tangent (tanh), sigmoid, Rectifier Linear Unit (ReLU).

$$y_i = \ell_i\left(\sum_j w_{ij}v_j\right) \quad (3.1)$$

The working principle of a neuron in the network is given in Figure 3.4 where $\ell_i(\cdot)$ is the activation function of the neuron i , w_{ij} is the weight from node j to node i . The output value of the neuron j is calculated as the weighted sum of the values of the neuron's input nodes is passed through the activation function (Equation 3.1).

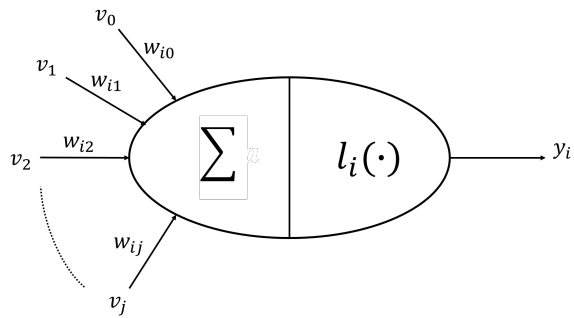


Figure 3.4: Working principle of a neuron

Feedforward networks and Backpropagation:

Figure 3.5 is a representative of a feedforward neural network, x_i is the i^{th} input and y_j is the j^{th} output. The input layer is the first layer of the network, takes input signals to pass them on the next layer without any operations applied to the input values. Hidden layers are the collection of neurons that apply different transformations to the data. The last hidden layer transfers the values to the output layer. Each neuron in a fully connected layer connects to every neuron in the next layer. The output layer is the last in the neural network, gives the prediction/classification result as output.

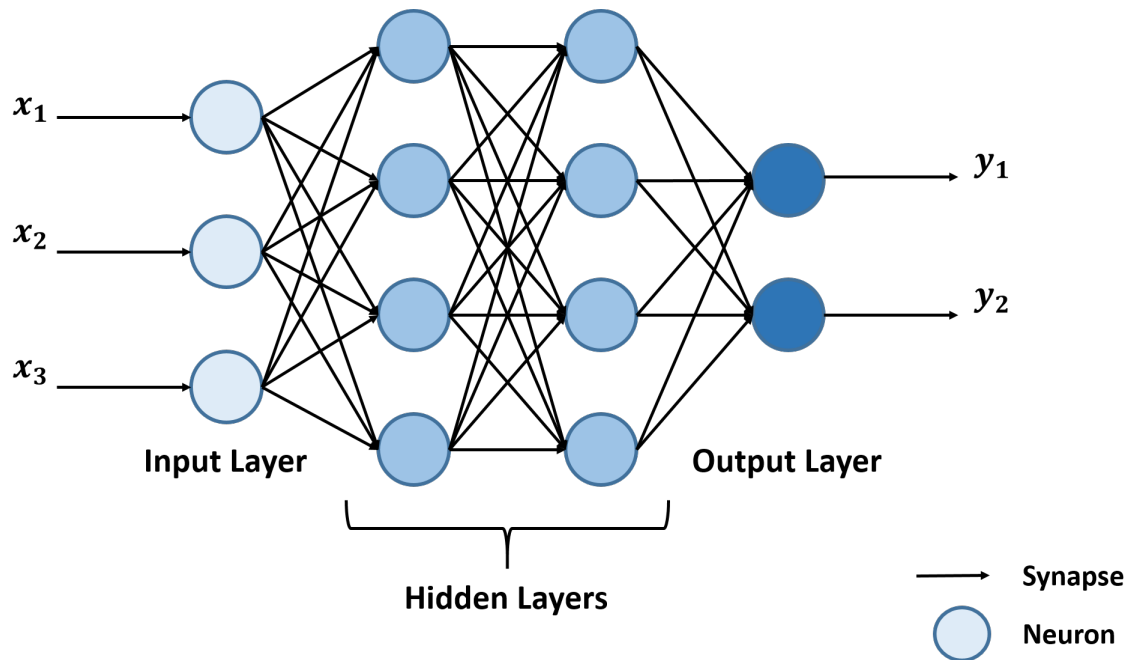


Figure 3.5: Feedforward Neural Network structure

Learning is a process performed by iteratively updating each one of the weights in a neural network to minimize the loss function that formulates the difference between desired and actual values for a sample of data. Input values are fed to the feed-forward neural network and the prediction value is obtained from the output layer. The output is then compared with the target value to calculate error by using a loss function.

Back-propagation is a training algorithm that uses the derivative of the loss function with respect to each parameter in the neural network [55]. For back-propagation, the derivative of the error is calculated by using the chain rule, then the derivatives are used to calculate the gradients of the other layers. This process is repeated until all gradients for each weight in the neural network is done. The learning rate is a hyperparameter used in optimization, adjusts the amount of the weight update. The gradients calculated by the back-propagation are multiplied by the learning rate and the multiplication is subtracted from the weight values to reduce the error.

Gradient descent is an optimization algorithm that iteratively updates the learnable parameters of the network to minimize the loss. The aim is to find the direction in which the loss function has the steepest rate of increase. Then, the parameters are updated in the negative direction of the gradient with a step size determined according to the learning rate.

The gradient update is calculated with Equation 3.2 where Δw_j is the weight update computed as in Equation 3.3:

$$w_j := w_j + \Delta w_j \quad (3.2)$$

$$\Delta w_j = -\eta \frac{\partial \ell}{\partial w_j} \quad (3.3)$$

Generally, stochastic gradient descent (SGD) method is used to train the neural networks by mini-batches, it is also called mini-batch gradient descent. Where n is defined as the mini-batch size, the average gradient of the loss function ℓ_i with respect to weight w_j is calculated by Equation 3.4.

$$\frac{\partial \ell}{\partial w_j} = \frac{\partial}{\partial w_j} \frac{1}{n} \sum_{i=1}^n \ell_i \quad (3.4)$$

The process steps to calculate the gradient in a feedforward neural network are as follows:

- i. Propagate a sample forward through the network to produce a value v_j at each neuron and output \hat{y} at the last layer.
- ii. Compute a loss function value $\ell(y, \hat{y})$ at each output node k .
- iii. Calculate δ_k for each output node k by the Equation 3.5:

$$\delta_k = \frac{\partial \ell(y_k, \hat{y})}{\partial \hat{y}} \ell'_k(\alpha_k) \quad (3.5)$$

- iv. Calculate δ_j for each node in the prior layer:

$$\delta_j = \ell'(\alpha_j) \sum \delta_k w_{kj} \quad (3.6)$$

δ_j represents the derivative of the total loss function concerning that node's incoming activation α_j , can be expressed as $\partial \ell / \partial \alpha_j$.

$$\frac{\partial \ell}{\partial w_{jj'}} = \delta_j v_{j'} \quad (3.7)$$

A CNN+LSTM deep learning model structure is given in Figure 3.6 where the feature maps obtained from CNN are input to the LSTM after flattening.

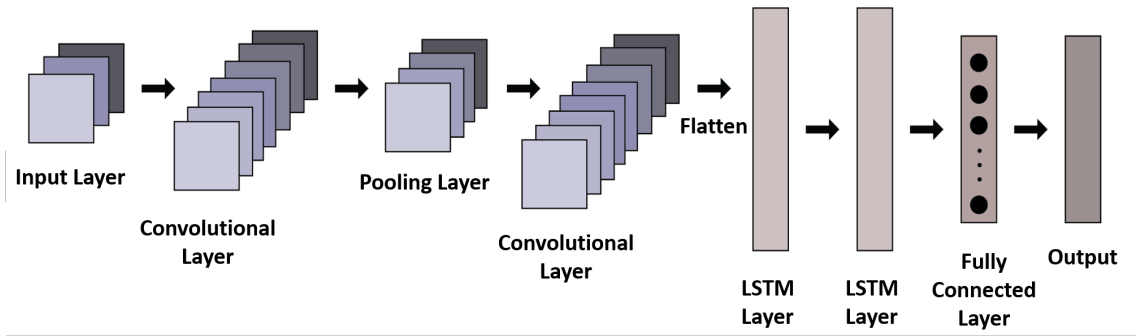


Figure 3.6: CNN+LSTM Deep Learning structure

3.3.1 Convolutional Neural Network (CNN)

Convolutional Neural Network is a type of Artificial Neural Networks and initially used to process image data. CNN is a category of deep learning that is the structure of a multilayer perceptron with more than one hidden layer. Hidden layers of a typical CNN consist of a convolutional layer, pooling layer, activation function, and flatten layer.

Convolutional Layer: Convolutional layer is a set of neurons that act like convolutional filters to generate feature maps. Input data are split into small blocks in this layer, and each block convolves with a specific set of weights. A different set of features are obtained by sliding convolutional filter on the input data with the same weights. Equation 3.8 displays the convolution operation:

$$F_l^k = (A * K_l^k) \quad (3.8)$$

where A is the input matrix, K_l^k represents the l^{th} convolutional filter of the k^{th} layer, F_l^k represents the output feature map of the convolutional layer.

Pooling Layer: Pooling is a local operation that reduces the spatial size of the representation to reduce the number of parameters. This layer determines the similar information in the local region and outputs the dominant response. Features that

are the output of the convolutional layer can occur at different locations in the image. Thus, their exact location becomes less important. The model uses the features' approximate position relative to others. Pooling provides generalization via extracting similar information and decrease the size of feature maps.

$$P_l = f_p(F^l) \quad (3.9)$$

The calculation for pooling is given in Equation 3.9, where F^l and P_l represent the l^{th} input and output feature maps and $f_p(.)$ is the pooling function.

Activation Function: Activation function is used for learning of complicated and nonlinear complex patterns. The calculation for activation function is defined in Equation 3.10 where $f_A(.)$ is a nonlinear activation function which is used at the output of convolutional layer, F_l^k , and T_l^k is the output of the k^{th} layer for l^{th} input.

$$T_l^k = f_A(F_l^k) \quad (3.10)$$

If a linear function is defined for activation, a linear calculation is made no matter how many neurons there are in the layer. Therefore, having a nonlinear activation function is an essential part of the neural network. Sigmoid is a nonlinear activation function (Equation 3.11) that is used where the output is desired between 0 and 1. The graph of the sigmoid function is given in Figure 3.7a.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (3.11)$$

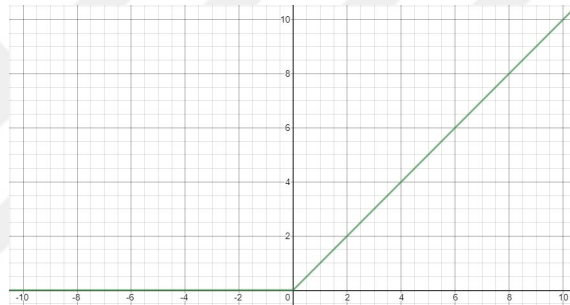
ReLU is a commonly used activation function in deep neural networks and it takes the range $[0, \infty)$. $f(z)$ is zero when z is less than zero and $f(z)$ is equal to z when

z is greater than or equal to zero (Equation 3.12). The graph of ReLU function is given in Figure 3.7b.

$$f(z) = \max(0, z) \quad (3.12)$$



The graph of the sigmoid activation function



The graph of the ReLU activation function

Figure 3.8: Two nonlinear activation functions used in neural networks

Flatten Layer: Flatten layer transforms the multi-dimensional array of features into a vector before the fully connected layer.

Fully Connected Layer: The fully connected layer analyzes the output of all previous layers globally by making a nonlinear combination of features, it is implemented at the end of the network for the classification/regression task.

3.3.2 Recurrent Neural Networks

Feed-forward networks only process information forward to remember what they have learned during training. On the other hand, in recurrent networks, they remember what they have learned from prior inputs. Thus, the result depends not

only on the current information but also on the previous information. Recurrent structures differ from feed-forward structures since they use the output as input in the next process. The connections between nodes are set to exhibit temporal dynamic behaviour. RNNs have stored states called gated state or gated memory that can be replaced by another network or graph (it is a part of LSTM).

The reason they have memory is that an input set coming in a specific order has a meaning for the output. Feed-forward networks are inadequate on this type of data (such as speech or handwritten text). In the RNNs, the output is influenced not only by the weights applied on inputs but also by a hidden state vector that represents the context based on prior inputs and outputs. Therefore, the network can produce different outputs by the same input that depends on the previous inputs in the series.

Recurrent Neural Network (RNN) has loops that allow information to pass from one step to the next. Figure 3.9 displays a loop of RNN with hidden state A , input x_t and output o_t , and an RNN structure that is made by unrolled loop.

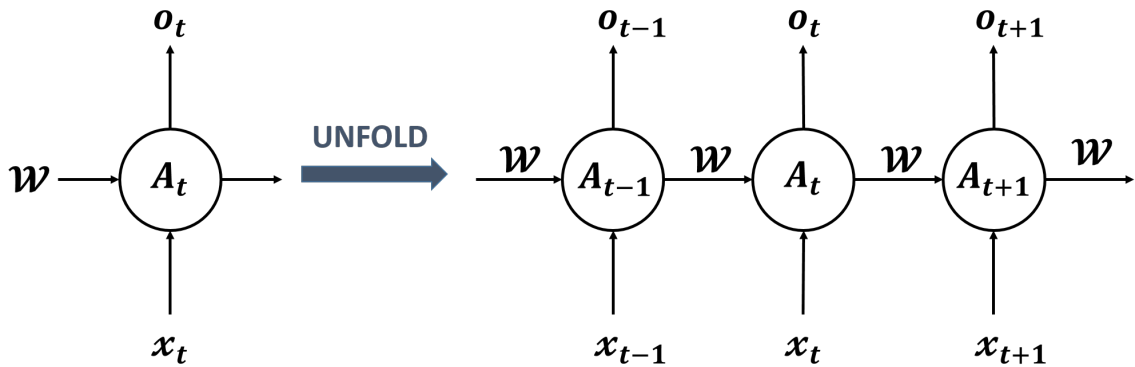


Figure 3.9: Recurrent Neural Network structure

x_t is input and o_t is the output at time step t , where A_t is the hidden state (or called as memory) of the network. A_t captures information about the sequence and performs its operation based on the previous state and the current input by using a nonlinear function. The output o_t and x_{t+1} are then used as the input for the next step $t + 1$. The network keeps remembering the context during training in this way.

Generally, Deep Neural Networks (DNNs) use different parameters at each layer. However, RNNs use the same parameters as weights (W) in all steps to perform the same task at each layer of the network with different inputs.

In RNN, the distance between the relevant information and the point where it is needed to make a prediction can be very wide [56]. Hidden state activations of RNN are affected by other local activations closest to them. That is the reason RNN has a short-term memory. In this situation, RNN cannot learn this long-term dependency since this problem causes vanishing/exploding gradient problem. Long Short-Term Memory (LSTM) is used to capture long-term dependencies in a sequence data.

3.3.3 Long Short-Term Memory

RNNs use one nonlinear function in each hidden state through the repeating network model. Unlike RNNs, LSTMs use more than one nonlinear function at the hidden state A to remember information for long periods and this more complex unit is called a memory cell.

The repeating module in an LSTM contains four interacting layers connected in a special way, which are also called as gate. Information received from outside can be stored, written to the memory cell, and read via these gates. These gates decide when information is stored, written, read, or deleted by the cell. The diagram in Figure 3.10 represents a basic RNN cell representation, predictions are generated via a combination of previous predictions and new information. Their units (called memory, selection, and ignoring) are added to this cell to make an LSTM cell model. Each unit has its neural network and nonlinear function.

In the second diagram (Figure 3.11), a memory is added to the cell to decide by using previous information for a long time. The signs of $+$ and \times express element-based addition and element-based multiplication. A copy of those predictions is held on by the memory unit. The multiplication gate decides which data and how much of it will be used by multiplying weights and the information in memory. While

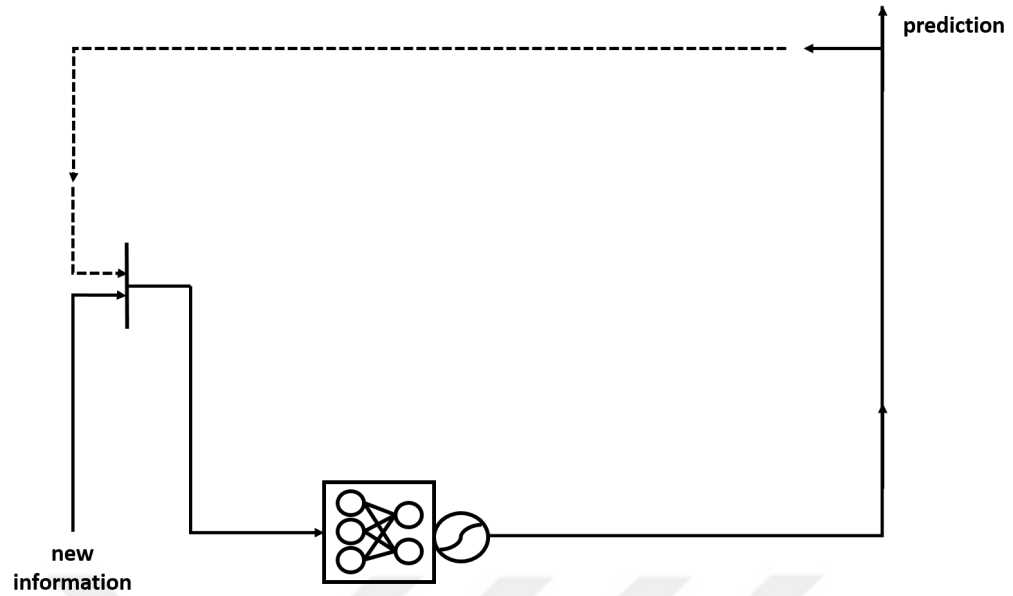


Figure 3.10: An RNN cell model

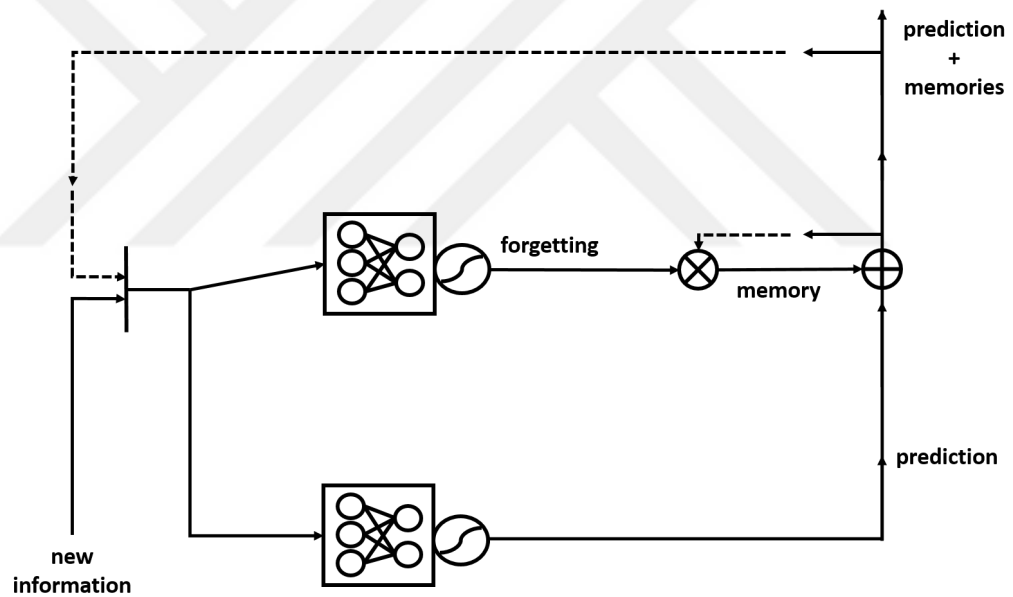


Figure 3.11: The RNN cell model with memory unit

some of them are forgotten, some of them are remembered and added back to the prediction.

In the third diagram (Figure 3.12), the selection unit is added to the cell and acts like a filter that keeps memories inside and let predictions to get out. New information and previous predictions are used to decide what should be kept internal and what

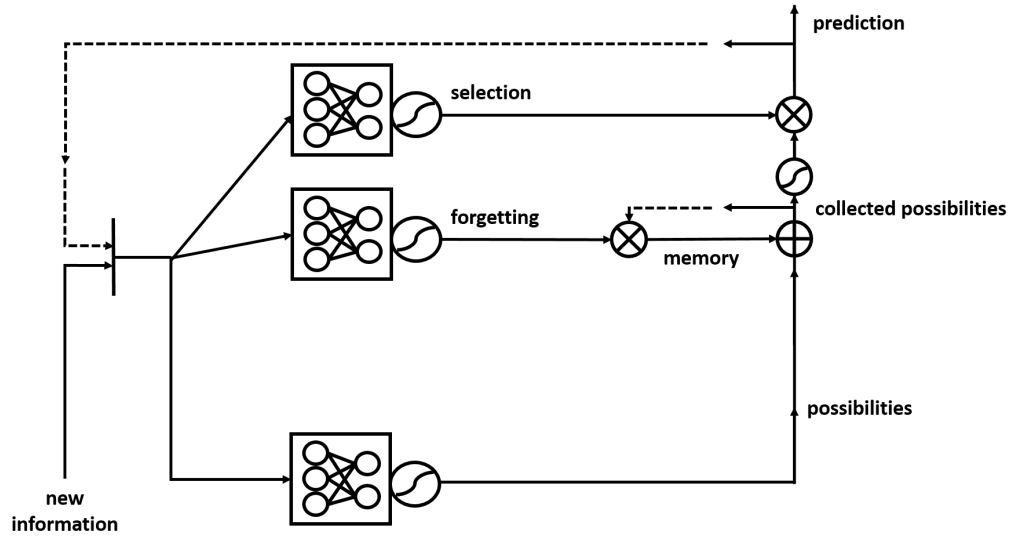


Figure 3.12: The RNN cell model with memory and selection units

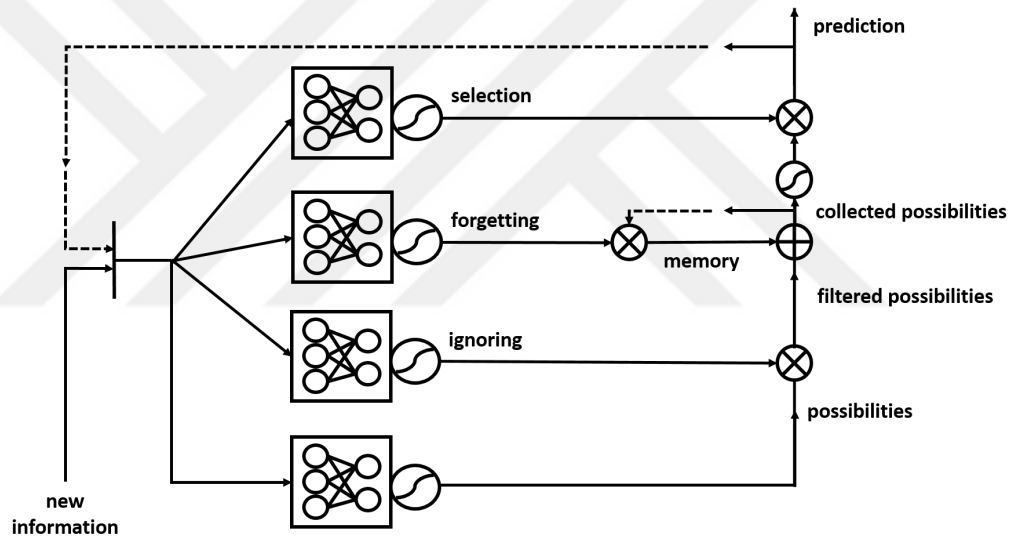


Figure 3.13: An LSTM cell model

should be released as a prediction. Thus, a few predictions are selected to be released as the prediction for that moment. The last unit, ignoring unit, is added in diagram-4 (Figure 3.13) to set apart information that is not relevant immediately, thus they cannot affect the predictions in memory that are going forward.

3.4 Transfer Learning

A domain $D = \{\chi, P(X)\}$ is defined by two parts:

- i. Feature space χ
- ii. Marginal probability distribution $P(X)$, where $X = \{x_1, x_2, \dots, x_n\} \in \chi$

where X denotes a particular learning sample where χ is the space of all possible feature vectors, and x_i is the i^{th} feature vector where n is the number of feature vectors in X . A task $T = \{Y, f(\cdot)\}$ is defined by two parts:

- i. A label space $Y = \{y_1, y_2, \dots, y_m\}$
- ii. An objective function $f(\cdot)$ which is not observed but can be learned from the training data by pairs $\{x_i, y_i\}$

where $f(\cdot)$ predicts the corresponding label $y_i \in Y$ of a new instance x , and $f(x)$ can be defined as $f(x) = P(y|x)$.

Within the scope of transfer learning, there are a source domain D_S and a target domain D_T , defined in the following equations.

$$D_S = \{(x_{S_1}, y_{S_1}), \dots, (x_{S_n}, y_{S_n})\} \quad (3.13)$$

$$D_T = \{(x_{T_1}, y_{T_1}), \dots, (x_{T_n}, y_{T_n})\} \quad (3.14)$$

The source instance and its corresponding class label are denoted by $x_{S_n} \in \chi_S$ and $y_{S_n} \in Y_S$. The target instance and its corresponding class label are given as $x_{T_n} \in \chi_T$ and $y_{T_n} \in Y_T$. The source task and the target task are denoted as T_S and T_T . The source prediction function and target prediction function are represented as $f_S(\cdot)$ and $f_T(\cdot)$.

Transfer learning aims to improve the learning of the target predictive function $f_T(\cdot)$

in given target domain D_T using the knowledge in source domain D_S and source task T_S where $D_S \neq D_T$ or $T_S \neq T_T$ where the target task is given as T_T .

The condition $D_S \neq D_T$ refers that either $\chi_S \neq \chi_T$ or $P_S(X) \neq P_T(X)$, and $T_S \neq T_T$ refers that either $Y_S \neq Y_T$ or $f_S(\cdot) \neq f_T(\cdot)$. If the target domain is same as source domain and their learning tasks are the same, the learning problem becomes a traditional machine learning problem [57].

Transfer learning can be classified into three categories [58]: (1) inductive transfer learning, (2) unsupervised transfer learning, and (3) transductive transfer learning. In inductive transfer learning the source task and target task are different from each other, ($T_S \neq T_T$). Unsupervised transfer learning is similar to inductive transfer learning but focuses on unsupervised learning tasks in the target domain. In transductive transfer learning source and target tasks are the same while the source domain is different but related to the target domain ($T_S = T_T, D_S \neq D_T$).

According to different relations between the source domain and target domain, transductive transfer learning is classified into two parts. (a) The feature space of the source domain is different from the target domain ($\chi_S \neq \chi_T$). (b) There is no difference in source and target domains but the marginal probability distributions of input data are different ($\chi_S = \chi_T, P_S(X) \neq P_T(X)$). Note that, in the case where $\chi_S = \chi_T$ it is called as homogeneous transfer learning, while the case where $\chi_S \neq \chi_T$ is defined as heterogeneous transfer learning [59].

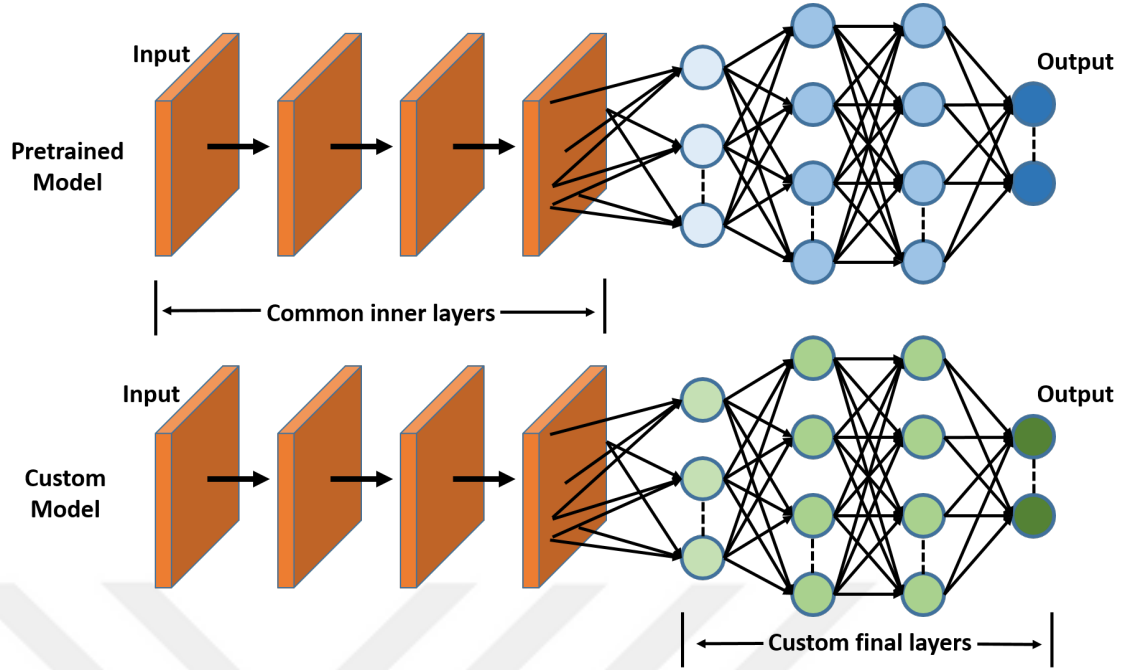


Figure 3.14: Representation of transfer learning between two neural network models

3.5 Performance Metrics

There are two common performance metrics used to evaluate the prediction performance of the system.

Root Mean Square Error (RMSE) is a common performance measure by calculating as following equation:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y - \hat{y})^2} \quad (3.15)$$

In Equation 3.15, n represents the number of instances, where y and \hat{y} are the actual and predicted values of the target feature respectively.

Mean Absolute Error (MAE) measures average magnitude of the errors without considering their direction. Equation 3.16 gives the calculation of MAE:

$$MAE = \frac{1}{n} \left(\sum_{i=1}^n |\hat{y}_i - y_i| \right) \quad (3.16)$$

\hat{y}_i and y_i are the prediction and actual values for the i^{th} hour, n is the number of measurements.



4. IMPLEMENTATION

The flowchart is shown in Figure 4.1 for the proposed air pollution prediction system in this thesis. Firstly, the data were preprocessed for cleaning and interpolation, and the interpolated data were normalized because of the variety of variables' range. Then, the interpolated time series data set was created according to the certain frame size values with different data separation methods. After the 2-dimensional and 3-dimensional input structures are defined as the CNN input, a deep learning-based time series prediction system has been implemented using the CNN+LSTM model by using MATLAB Deep Network Designer Tool. Hyperparameter tuning was performed to increase the prediction performance of the model, the model was applied to the data of the three cities with various hyperparameter values to obtain the optimal network properties. Lastly, the pre-trained neural network, which had been trained with the previous data of a city, was run with transfer learning on another city's data that has a different distribution. RMSE and correlation coefficient between target and predicted outputs are measured to evaluate the prediction performance.

The model hyperparameters were optimized during the training and test processes. After deciding on neural network properties and hyperparameter values, the model was run 15 times for each method to select the neural network that reaches the lowest validation RMSE to use on test data. RMSE and correlation coefficient for the test were measured to evaluate the prediction performance of the model. If the test RMSE value dropped to the desired level, the results were simulated. Otherwise, the network was redesigned and trained by changing properties and hyperparameters such as hidden layer type and number, neuron number, activation function.

Method-1: A 2-dimensional array containing the information of target pollutant's concentration in past hours is input to the prediction model to obtain a predicted value for the concentration of that pollutant in the next hour.

Method-2: A 3-dimensional array containing the information of all pollutants' concentration and meteorological information in past hours is input to the prediction model to obtain a predicted value for the concentration of the target pollutant in the next hour.

Method-3: A 3-dimensional array containing the information of all pollutants' concentration and meteorological information in past hours is input to the prediction model to obtain predicted values for the concentration of all pollutants in the next hour.

It is mentioned that all methods output the predicted concentration of target pollutants in all locations at the same time.

Firstly, Method-1 was employed for non-normalized Barcelona data to predict the actual concentration value of the target pollutant. Then it was performed for the normalized data to compare the performance with the results of non-normalized data. Later, all three methods were performed with normalized data for each city. Besides, both air quality data and meteorological data were used for Kocaeli and İstanbul (in Method-2 and Method-3) to select the appropriate features for the best prediction performance. Therefore, temperature, relative humidity, air pressure, wind speed, and wind direction were added to the input in Method-2 and Method-3 along with the concentration of pollutants. However, meteorological data could not be used in the studies since there is only air pollution information in Barcelona data. For transfer learning, deep neural networks that were successful with Barcelona and Kocaeli data were tested on İstanbul data by transferring weights. Also, transfer learning performance has been evaluated in cases where different pollutants' information exist in the target domain and source domain (for example, between Barcelona and Kocaeli).

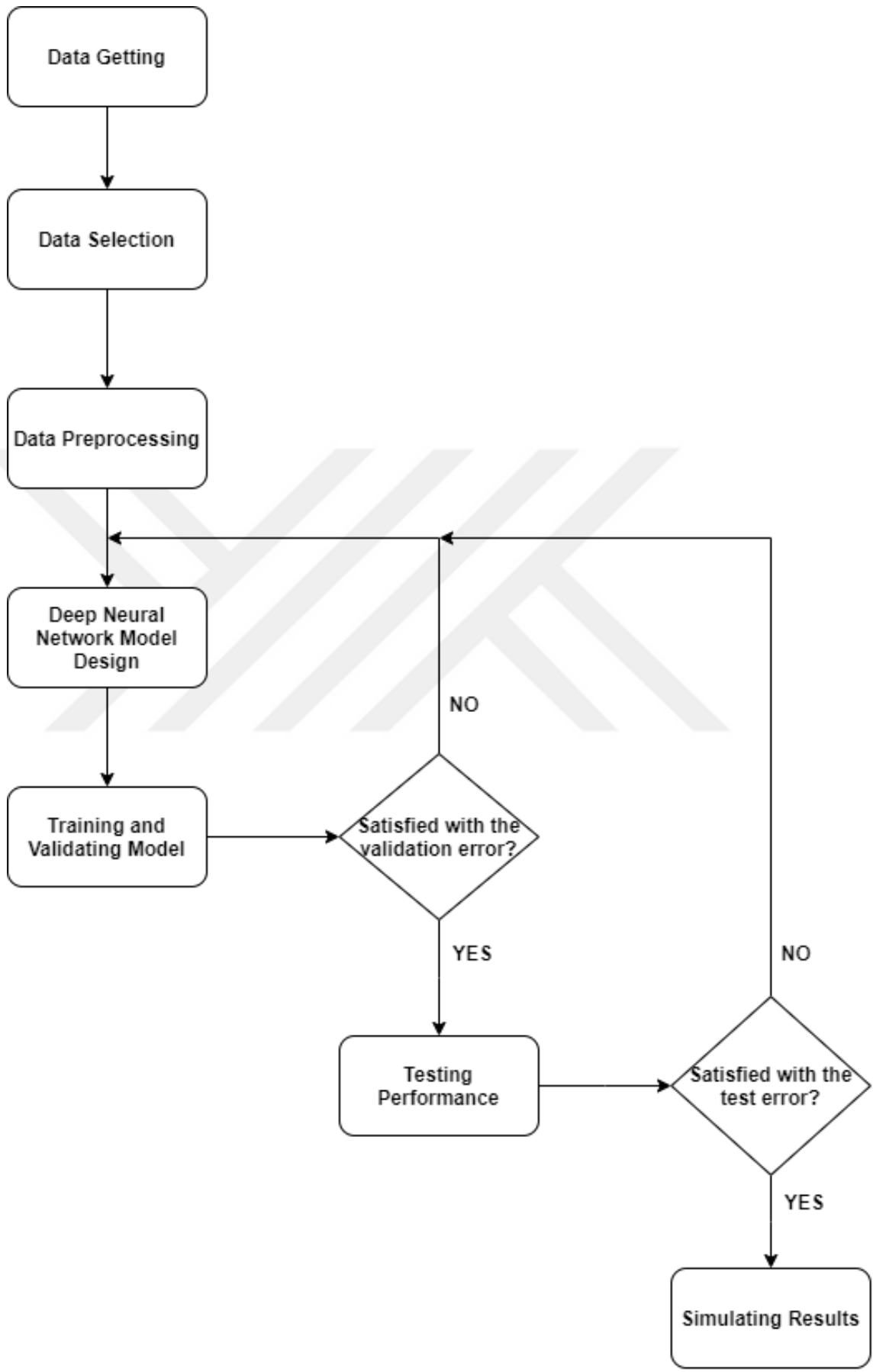


Figure 4.1: Flowchart of proposed air pollution prediction system

4.1 Data Preprocessing

4.1.1 Linear Interpolation

Linear interpolation is a curve fitting method to find a value between two points on a curve by using linear polynomials. There are two known value pairs (x_1, y_1) and (x_2, y_2) , the point (x_0, y_0) between those two points is specified with Equation 4.1:

$$y_0 = \frac{(y_2 - y_1)}{(x_2 - x_1)}(x_0 - x_1) + y_1 \quad (4.1)$$

In the data preprocessing phase, all duplicate data were cleaned, then linear interpolation was applied in places where there was missing values for at most five consecutive points. If the number of consecutive missing points were more than five, they are left as missing. A missing value rate was decided as a threshold to determine which locations were going to be used for pollutant concentration prediction. The histograms are given in Figures 4.2, 4.3 and 4.4 and the time series graphs are given in Figures 4.5, 4.6 and 4.7 for Barcelona.

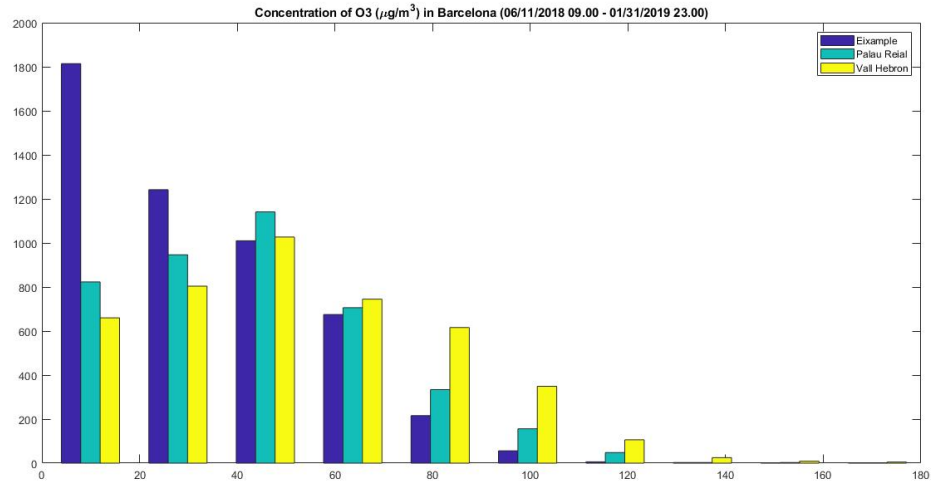


Figure 4.2: Histogram for the concentration of O_3 in Barcelona

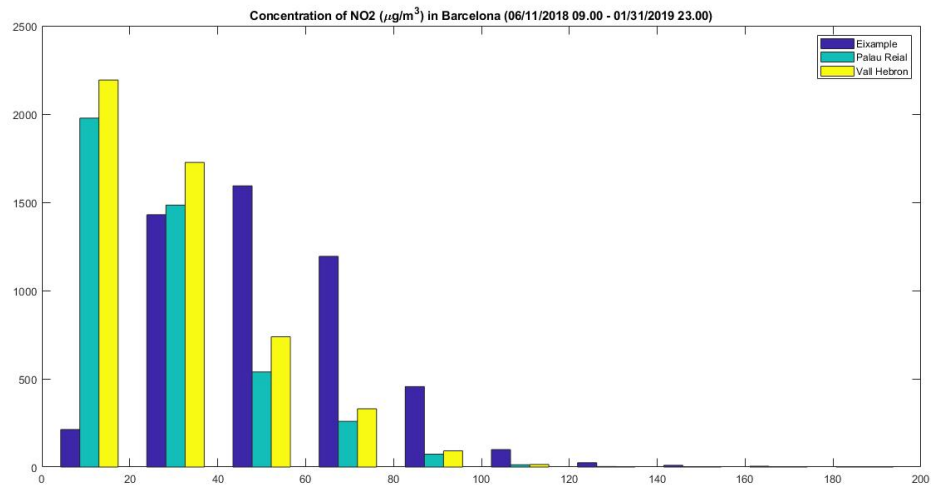


Figure 4.3: Histogram for the concentration of NO_2 in Barcelona

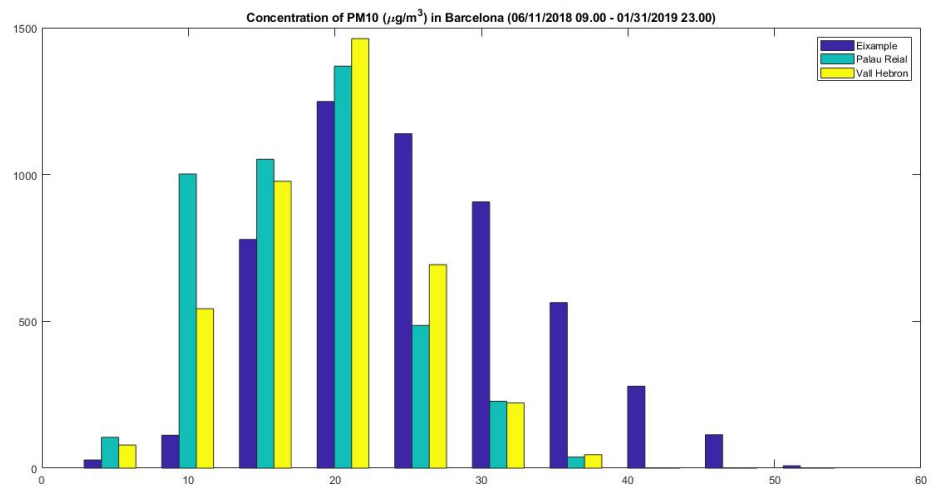


Figure 4.4: Histogram for the concentration of PM_{10} in Barcelona

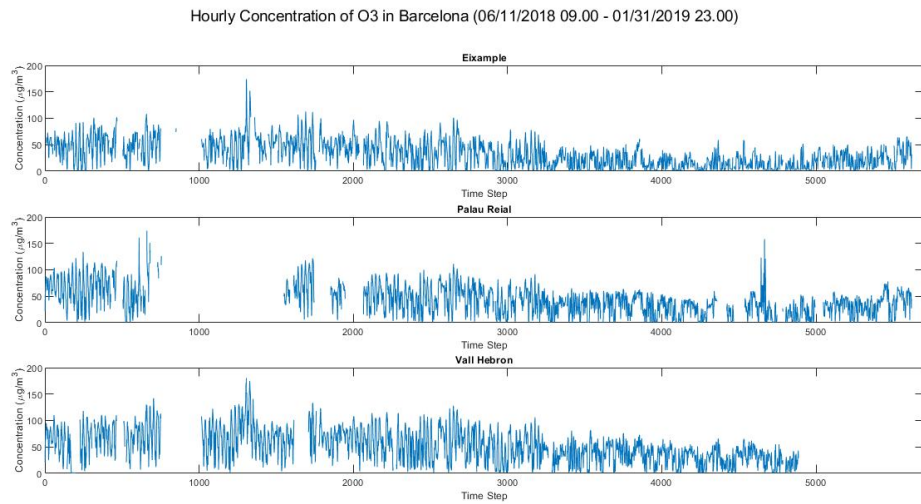


Figure 4.5: Hourly concentration of O_3 in Barcelona

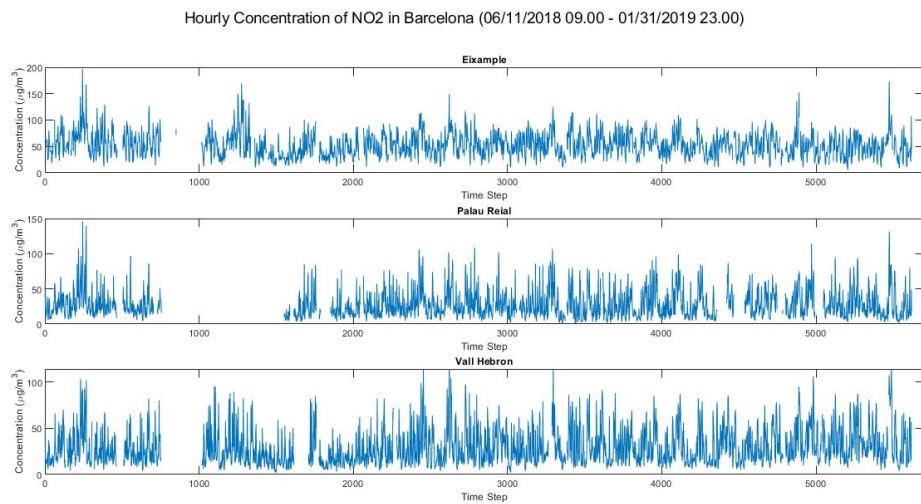


Figure 4.6: Hourly concentration of NO_2 in Barcelona

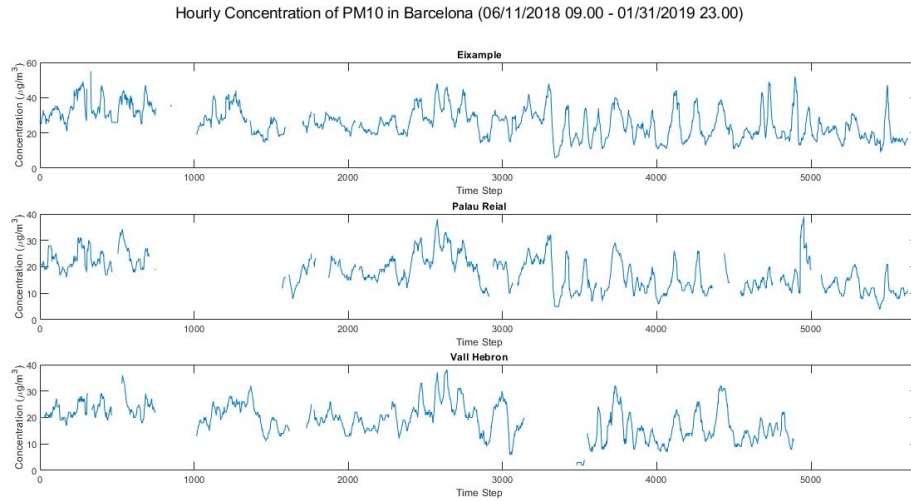


Figure 4.7: Hourly concentration of PM_{10} in Barcelona

4.1.2 Data Normalization

Min-Max Normalization:

The redundancy effect is that the features with very large values relative to other features have a big numerical effect on weight update during the training of the network. It has a negative impact on the metrics used to measure prediction success performance after testing. In order to solve these problems, the data were normalized with the min-max normalization method and time series input-output data sets were created with the normalized data.

Min-max normalization is a robust technique that converts data to a common range. Based on the min-max normalization, the smallest and largest values become equal to 0 and 1 respectively, and other values in the data set are spread to the interval $[0, 1]$ by Equation 4.2:

$$k' = \frac{(k - \min(k))}{\max(k) - \min(k)} \quad (4.2)$$

Where k is the original value and k' is the normalized value, $\max(k)$ and $\min(k)$ are also the maximum and the minimum values in the data set respectively.

4.2 Hyperparameter Tuning

Hyperparameter is a variable that is set before applying the learning algorithm to the data. Hyperparameters can be related to model selection tasks such as topology and size of the network, or the optimization and training process such as the learning rate. Hyperparameters and their domain/range are given as follows.

Frame Size: Frame size is taken as integer values in the interval $[8, 15]$.

Step Size: The step size of the time series sample is taken as 1 in this study.

Data Separation: Data can be split by 70%-15%-15% or 80%-10%-10% for training, validation, and test data sets.

Sample Selection: There are three sample selection methods: random, sequential and consecutive. Random selection means that when the data split into sets, training, validation and test samples are selected randomly from the data. Figure 4.8 shows the sequential and consecutive selection methods. Sequential validation type refers to the selection of validation samples sequentially from the training data set by using validation frequency where the data split into training and test consecutively. As for consecutive validation type, the data are split into three pieces for training, validation and test respectively.

Validation Frequency: Validation frequency is the number of iterations between evaluations of validation metrics and taken as 10, 15 and 20 respectively. It also refers to the period of selecting the validation sample from the training set when the sample selection type is sequential.

Mini-batch Size: Mini-batch size per iteration is taken as 30, 70, 100, 150 and

200 for training progress according to the amount of the time series data set.

Number of the epoch: In this study, the early stopping method is used to determine when to stop training the model. The training process is stopped if the validation error has not improved in the previous consecutive epochs.

Number of Convolutional Layer: The CNN part of the model is constructed with one, two and three convolutional layers respectively.

Pooling Layer: The CNN part of the network is built without a pooling layer, with an average pooling layer, and with a max-pooling layer.

Filter Size: In the CNN part of the model, the filters with size 2x2 and 3x3 are used.

Number of Filters: In the CNN part of the model, the number of filters is taken as 4, 5, and 10.

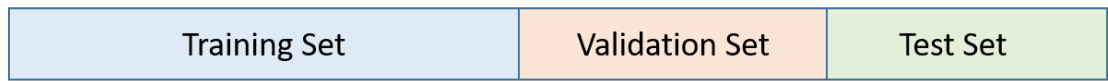
Number of LSTM Layer: The LSTM part of the model is constructed with one and two LSTM layers.

Number of Hidden Units in LSTM Layer: The number of hidden units is taken as 25, 50, 75, 100, 150 and 200.

Activation Function: Rectifier Linear Unit (ReLU), hyperbolic tangent (tanh) and sigmoid are used as activation functions.

Learning Rate: 0.001, 0.005, 0.01 are taken as learning rate.

Consecutive Selection



Sequential Selection

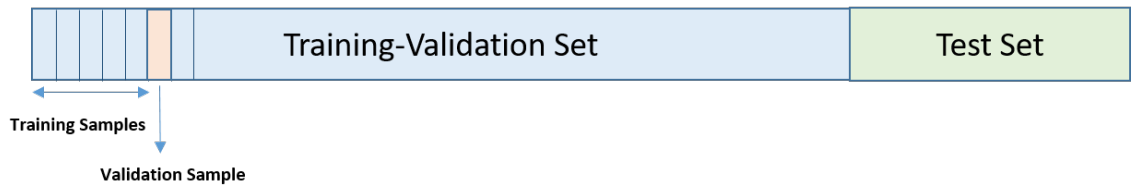


Figure 4.8: Consecutive and sequential sample selection types

5. RESULTS

A deep neural network structure has been developed to predict future concentrations of air pollutants by using pollutant concentration and meteorological information from past hours. For this purpose, the layers and their properties (the number of layers, the number of the filter, filter size, type of pooling, type of activation function, the number of the hidden neuron) with the best RMSE results were determined. The model structure of the proposed CNN+LSTM deep learning prediction model and its properties are given in Table 5.1 where the parameters are as follows:

- m : number of locations
- n : number of time steps
- k : number of features
- n_k : number of convolutional filters
- n_l : number of LSTM units
- n_o : number of neurons in fully connected layer.

Also, adding a pooling layer did not improve the success of the model. Hence, it caused a higher RMSE and unstable training performance. Since the pooling layer is generally used for a huge input data structure (such as an image with high resolution) to reduce the information to select the representative features, it loses the main features of a smaller input data.

The deep learning model was run 15 times for all combinations of hyperparameter values to predict the pollutant concentration by each method. Then the best results were chosen according to the lowest RMSE values, test RMSE and the correlation coefficient were measured. As a result, the average of test RMSE with the random

Table 5.1: Proposed deep learning model's layers and their properties

Layer Number	Layer Name	Layer Type	Properties
1	input layer	Sequence Input	Sequence input with $m \times n \times k$ dimensions
2	fold	Sequence Folding	Sequence folding
3	conv_1	Convolution	n_k 2x2 convolutions with stride [1 1] and padding 'same'
4	activation_1	ReLU	ReLU
5	conv_2	Convolution	n_k 2x2 convolutions with stride [1 1] and padding 'same'
6	activation_2	ReLU	ReLU
7	unfold	Sequence Unfolding	Sequence unfolding
8	flatten	Flatten	Flatten
9	lstm_1	LSTM	LSTM with n_l hidden units
10	activation_3	ReLU	ReLU
11	fc	Fully Connected	n_o fully connected layer
12	activation_4	ReLU, Sigmoid	ReLU, Sigmoid
13	regression	Regression Output	mean-squared-error

selection method is lower than the results with consecutive and sequential sample selection methods. Also, the learning rate was taken as 0.005 because it generally reached the lowest RMSE.

The main purpose of using ReLU after the fully connected layer is to eliminate negative values since the concentration of pollutants in the air cannot be negative. When Method-1 was performed for prediction with non-normalized data by using ReLU, the correlation between the target and predicted values are better than the results done with normalized data. However, ReLU causes prediction result to exceed target value. To prevent this problem, the sigmoid activation function was

used at the end of the fully connected layer, thus the prediction results were kept in the range $[0,1]$. Although the use of sigmoid function in the last layer could not capture the peak points in the time series prediction, the results are better than results with ReLU. Moreover, the neural network was run by setting the activation function of each hidden layer as the sigmoid. Results show that using sigmoid in each layer causes the prediction model to fail. In conclusion, compared to other pollutants, the model has lower test RMSE for the PM_{10} concentration in Barcelona, Kocaeli, and İstanbul.

5.1 Method-1

In order to evaluate the results of the Method-1, 2D samples from non-normalized Barcelona data were produced and each one's historical information was used to predict the concentration in $\mu g/m^3$. The neural network with the best validation RMSE value was tested and performance metrics were calculated. The test RMSE values are 37.29, 29.58, and 13.87 for O_3 , NO_2 and PM_{10} respectively while the correlation coefficients are 0.87, 0.79, and 0.96. A scatter plot for the prediction result of PM_{10} concentration with the best performance is in Figure 5.1, the graph of the predicted and target values are given in Figure 5.2.

In the second stage, the data of Barcelona was used for normalized values, and the same process was repeated for the new data sets, results with using ReLU and sigmoid after fully connected layer is shown in Figure 5.3 and Figure 5.4.

Because of the normalized data by the min-max normalization method, the target values are between 0 and 1. Although using the ReLU activation function after the fully connected layer eliminated the negative values resulting from prediction, it continued to hold high values. Therefore, when the predicted value is higher than the target value, ReLU holds this exceeding value. On the other hand, sigmoid limits the prediction values obtained after the fully connected layer to the range $[0, 1]$. In order to solve the problem of ReLU, sigmoid is set as the activation function

Test Results of Method-1 for PM10 Concentration in Barcelona with non-Normalized Data

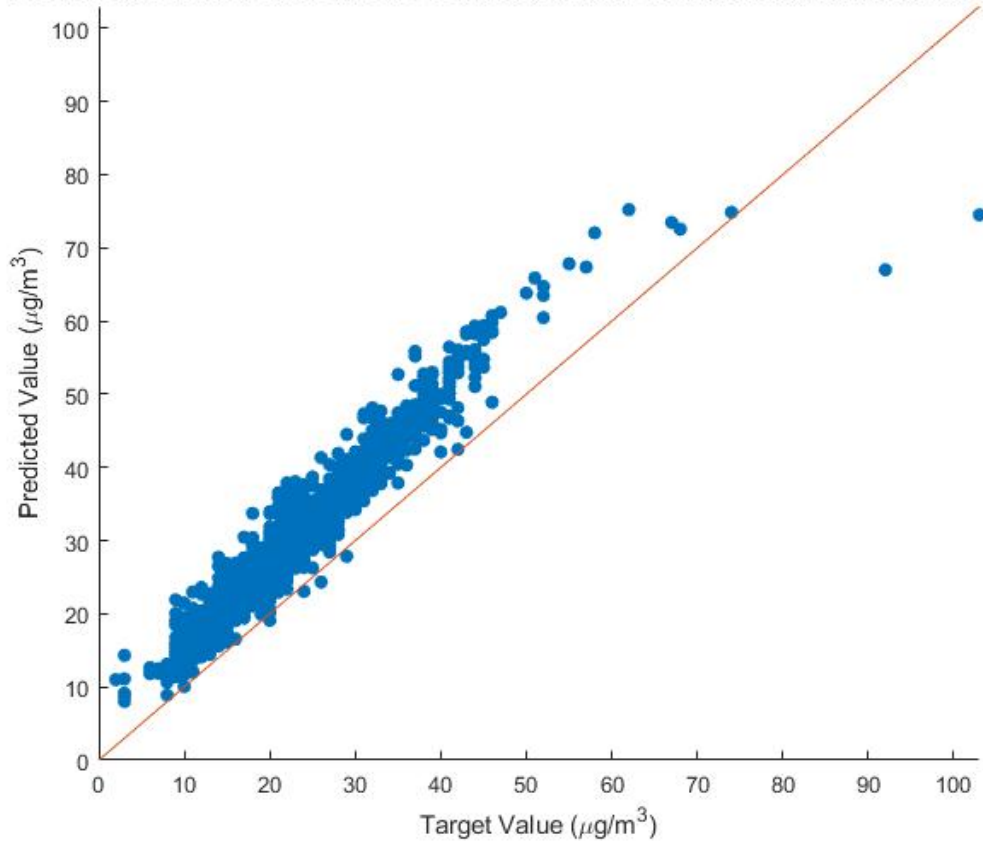


Figure 5.1: Results of Method-1 for the prediction of PM_{10} concentration in Barcelona (with non-normalized air pollution data)

Target-Predicted Concentration of PM10 in Barcelona with non-Normalized Data (by Method-1)

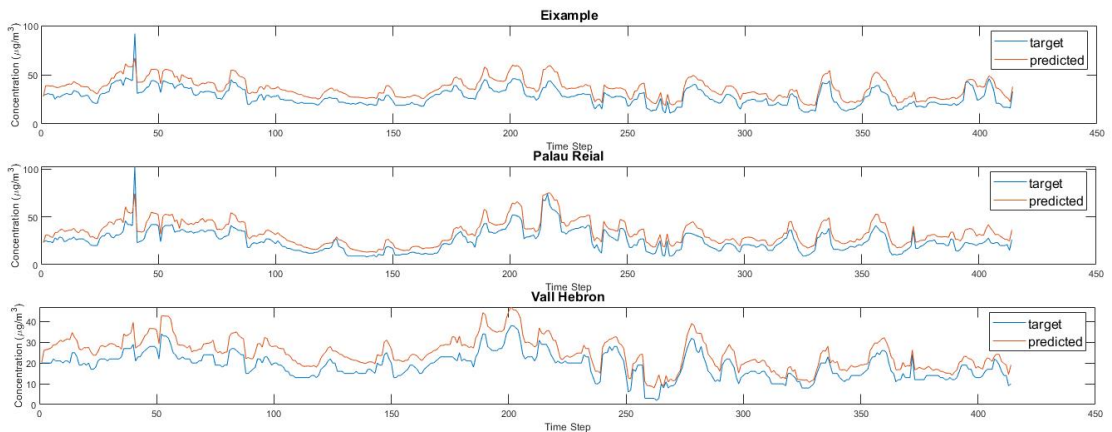


Figure 5.2: Graph of the predicted and target values for the prediction of PM_{10} concentration in Barcelona (with non-normalized air pollution data) by Method-1

of the last learning layer in the designed neural network. Sigmoid gave the best performance metrics values.

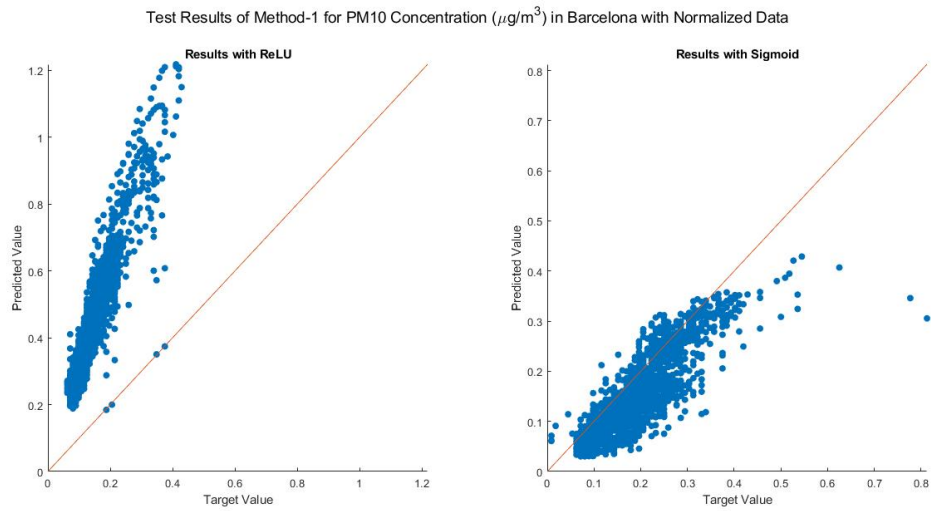


Figure 5.3: Results of Method-1 for the prediction of PM_{10} concentration in Barcelona (with normalized air pollution data)

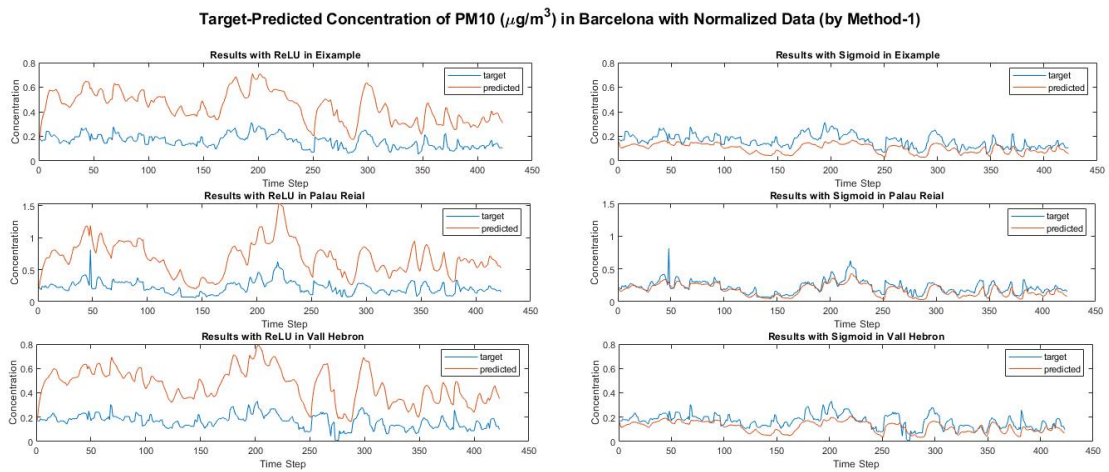


Figure 5.4: Graph of the predicted and target values for the prediction of PM_{10} concentration in Barcelona (with normalized air pollution data) by Method-1

The test RMSE with sigmoid was obtained as 0.06, 0.09 and 0.06 for O_3 , NO_2 and PM_{10} respectively where the test RMSE with ReLU was observed as 0.28, 0.39 and 0.42.

5.2 Method-2

3D samples were created from normalized data to predict the concentration of one target pollutant in the next hour by using the historical information of all pollutants in a city's data. The pollutants have different concentration value ranges and the normalization prevents the effect of numbers that are too large compared to the others during weight update. Also, since sigmoid's ability to limit the prediction values to a range between 0 and 1, the use of sigmoid instead of ReLU in the fully connected layer decreases the test RMSE. The test RMSE was calculated as 0.07 for Barcelona and 0.09 for Kocaeli, the relationship between target and predicted values are shown in Figure 5.5 for Barcelona and Figure 5.7 for Kocaeli.

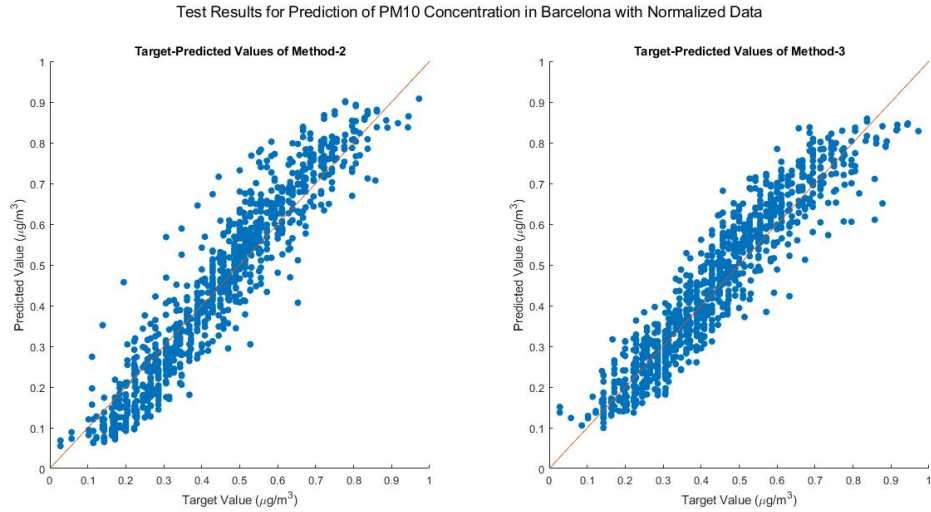


Figure 5.5: Scatter plot of target and predicted concentrations for PM_{10} in Barcelona (with air pollution data) by Method-2 and Method-3

İstanbul data consist of information about five pollutant's concentration (O_3 , NO_2 , PM_{10} , SO_2 , NO_X). To compare the prediction performance of the proposed deep learning model in three cities, two different time series data sets were prepared from İstanbul data. While, the first time series data set was created from the information of the same pollutants with Barcelona (O_3 , NO_2 , PM_{10}), the second data set was created by using the information of the same pollutants with Kocaeli (PM_{10} , SO_2 , NO_X). The test RMSE values for Dataset-1 and Dataset-2 in İstanbul were measured as 0.08 and 0.09 respectively. The relationship between target and

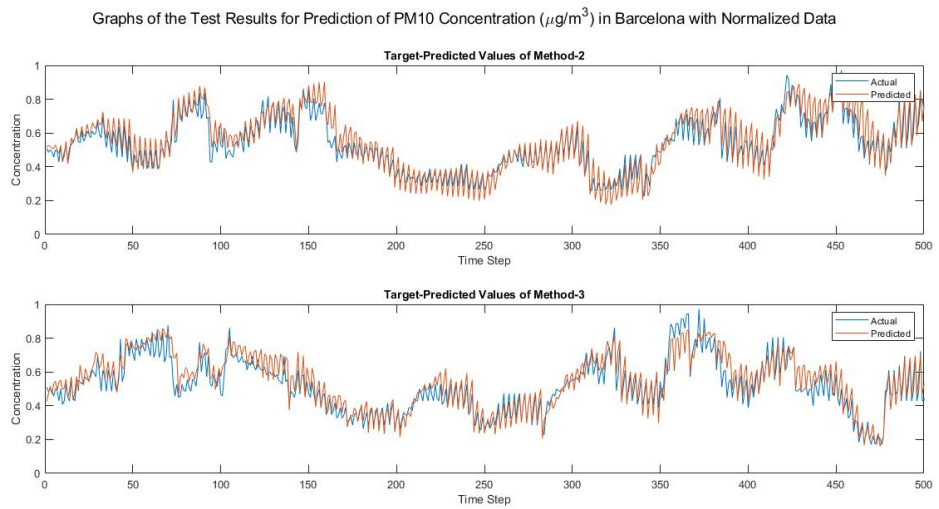


Figure 5.6: Graph of the predicted and target values for the prediction of PM_{10} concentration in Barcelona (with normalized air pollution data) by Method-2 and Method-3

predicted values in İstanbul is shown in Figure 5.9 for Dataset-1 and in Figure 5.11 for Dataset-2.

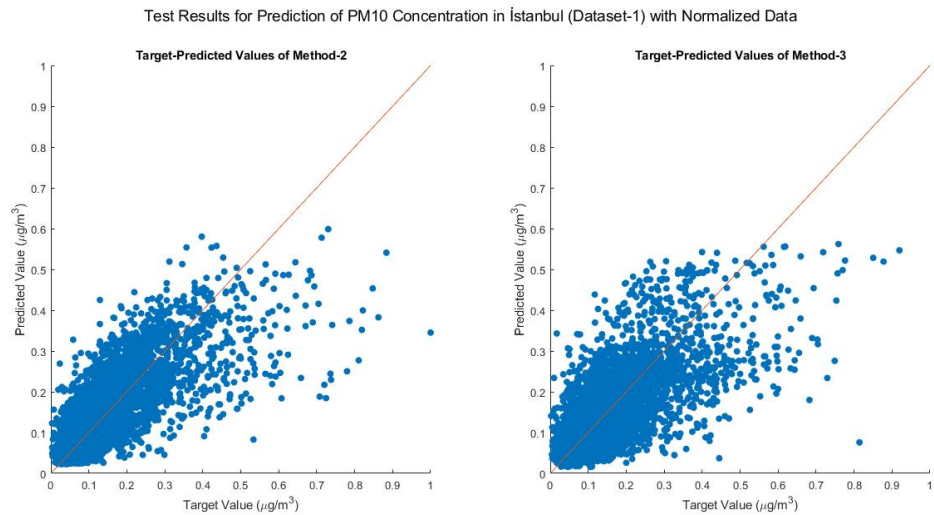


Figure 5.9: Scatter plot of target and predicted concentrations for PM_{10} in İstanbul (with air pollution data) by Method-2 and Method-3 (Dataset-1)

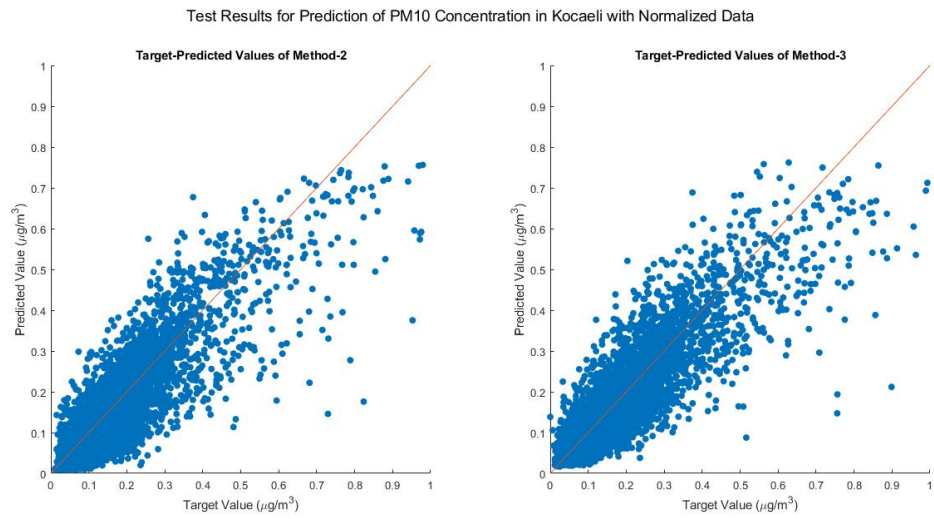


Figure 5.7: Scatter plot of target and predicted concentrations for PM_{10} in Kocaeli (with air pollution data) by Method-2 and Method-3

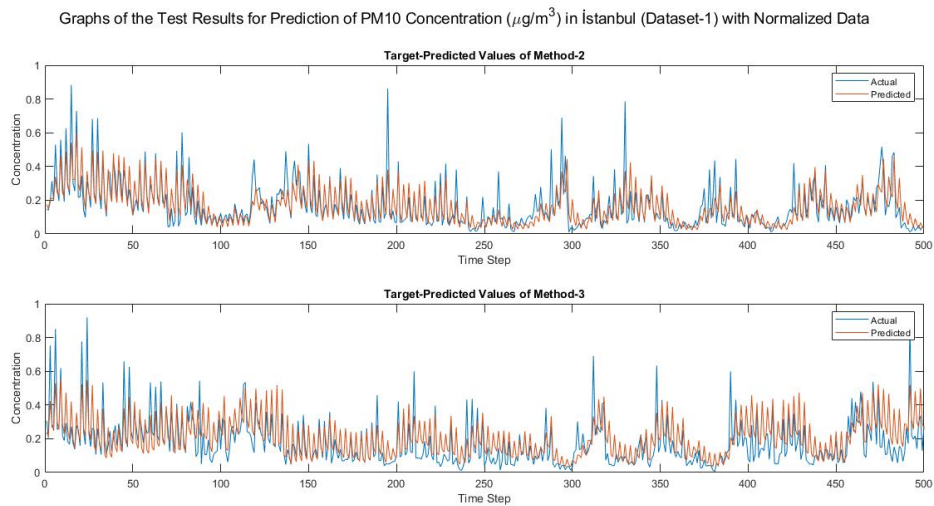


Figure 5.10: Graph of the predicted and target values for the prediction of PM_{10} concentration in İstanbul (with Dataset-1) by Method-2 and Method-3

5.3 Method-3

Method-3 was performed to predict the concentration of all pollutants at the same time in a city. The input contains the historical information of all air pollutants in all locations, and the target is the concentration of all pollutants in the next hour. 3D samples created with only normalized pollutant concentrations were input to

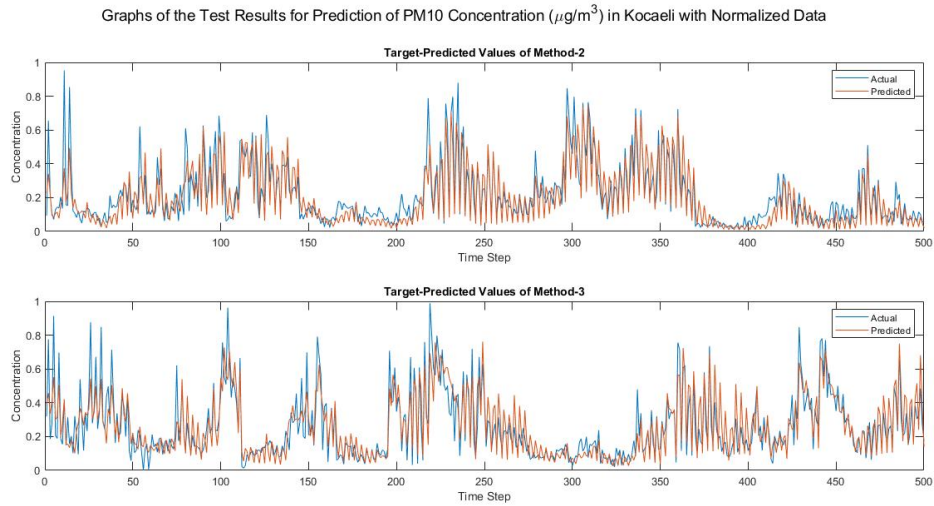


Figure 5.8: Graph of the predicted and target values for the prediction of PM_{10} concentration in Kocaeli (with normalized air pollution data) by Method-2 and Method-3

the system. Sigmoid is used instead of ReLU as the activation function in the fully connected layer since it provides a decrease in the test RMSE due to its ability to limit the prediction values to a range between 0 and 1.

Where test RMSE in Barcelona was calculated 0.07, the test RMSE in Kocaeli was calculated as 0.10 for Method-3. The relationship between target and predicted values are shown in Figure 5.5 for Barcelona and Figure 5.7 for Kocaeli. The graph of the target-predicted values are given in Figure 5.6 for Barcelona and Figure 5.8 for Kocaeli.

As for the results of İstanbul, the test RMSE was measured as 0.09 for Dataset-1 and 0.10 for Dataset-2. The relationship between target and predicted values in İstanbul is shown in Figure 5.9 for Dataset-1 and Figure 5.11 for Dataset-2. The graph of predicted-target values in İstanbul are shown in Figure 5.10 for Dataset-1 and Figure 5.12 for Dataset-2.

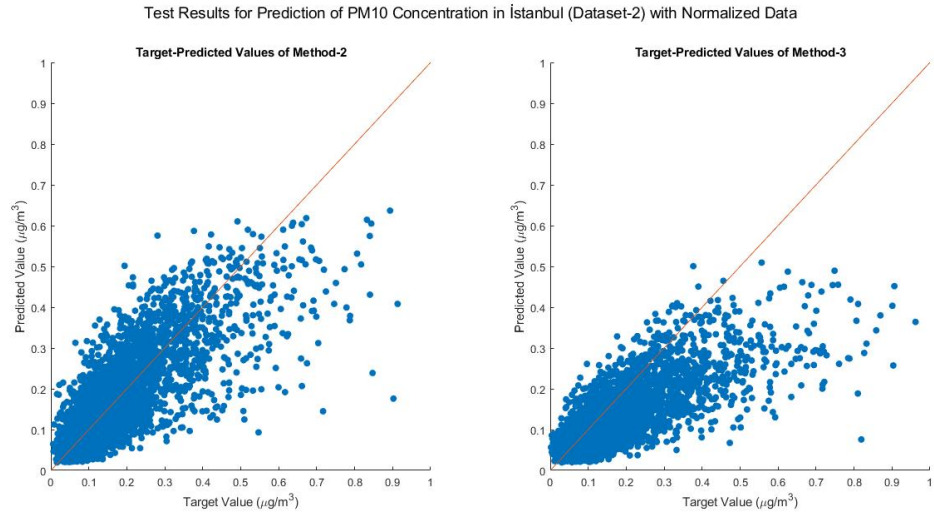


Figure 5.11: Scatter plot of target and predicted concentrations for PM_{10} in İstanbul (with air pollution data) by Method-2 and Method-3 (Dataset-2)

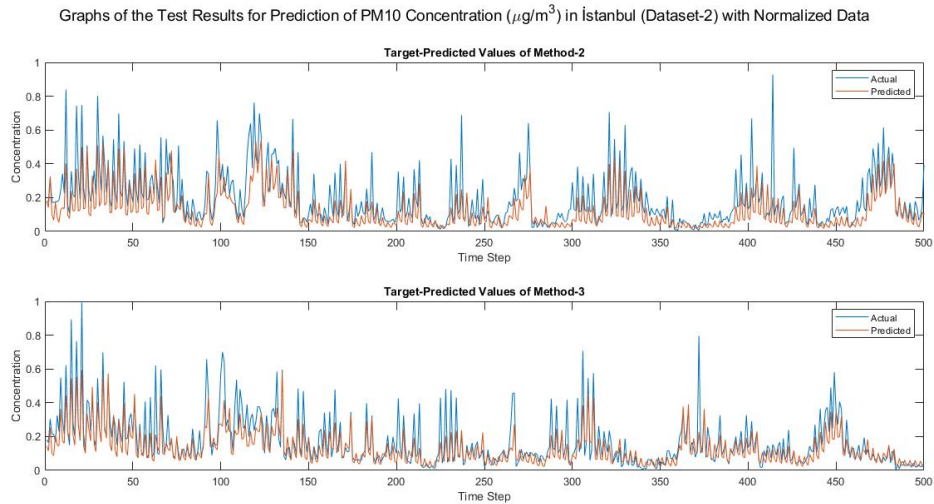


Figure 5.12: Graph of the predicted and target values for the prediction of PM_{10} concentration in İstanbul (with Dataset-2) by Method-2 and Method-3

Results with Meteorological Information

To observe the contribution of meteorological information to the prediction performance, past meteorological information such as temperature, relative humidity, air pressure, wind speed, and wind direction were added to the input as features. CNN+LSTM deep learning model was used for prediction, the results were obtained

with sigmoid after fully connected layer. Meteorological data were used for Kocaeli and İstanbul since they are available for only these two cities. The results given in Table 5.2 were observed when sigmoid was used as the activation function for the last learning layer. For the prediction of PM_{10} concentration, using meteorological information (MI) data with air quality (AQ) data reduced the RMSE value by 10% with Method-2 and Method-3 in İstanbul. However, it increased RMSE by 12% with Method-2 and 30% with Method-3 in Kocaeli. For the prediction of SO_2 , adding MI to the input caused a 15% increase with Method-2 but a 10% decrease with Method-3 in Kocaeli. On the other hand, there was any change in the test RMSE in İstanbul. As for the prediction of NO_x concentration, the use of MI data with AQ data caused an increase in both two cities. The RMSE increased by about 45% in İstanbul and 25% in Kocaeli with Method-2; it increased by 33% in Kocaeli and 10% in İstanbul with Method-3.

Table 5.2: Test RMSE results for Kocaeli and İstanbul with air quality (AQ) and meteorological information (MI) data

Pollutant	Study	Kocaeli		İstanbul	
		with AQ Data	with AQ+MI Data	with AQ Data	with AQ+MI Data
PM_{10}	Method-2	0.08	0.09	0.08	0.07
	Method-3	0.07	0.09	0.09	0.08
SO_2	Method-2	0.07	0.08	0.09	0.09
	Method-3	0.09	0.08	0.10	0.10
NO_x	Method-2	0.08	0.10	0.09	0.13
	Method-3	0.09	0.12	0.10	0.11

5.4 Transfer Learning

The source domain and target domain were determined to create a deep neural network model supported by transfer learning. Transductive transfer learning was selected for the transfer of the weights of the pre-trained CNN+LSTM model. In transductive transfer learning, the source task and target task are the same, the source domain and target domain are different but related. Since the model has better prediction performance for PM_{10} concentration than other pollutants in all

cities according to the test RMSE, transfer learning was used to predict PM_{10} concentration by methods Method-2 and Method-3. Barcelona and Kocaeli were selected as the source domain where the weights transfer from, to obtain a more accurate prediction system on İstanbul data, then weight transfer was done between Barcelona and Kocaeli.

Firstly, pre-trained networks that had been trained with the source domain were applied to target domains without any training process with the training data set of the target domain. Then, the fully connected layer, which is the last learning layer of the pre-trained network, was replaced with a new one. The network with the new fully connected layer was trained with the target city's training data. The learning rate was set as 0.001 for the training of transfer learning training.

The graphs in Figure 5.14 show the predicted and target values for PM_{10} concentration in İstanbul with transfer learning, and Figure 5.13 shows the relationship of target-predicted values with transfer learning results. Kocaeli data are selected as the source domain and Dataset-2 for İstanbul is selected as the target domain. Test RMSE values of transfer learning for Method-2 and Method-3 were calculated as 0.12 and 0.09 respectively.

There are three main results from studies with transfer learning. Firstly, testing pre-trained neural networks on the target domain without any training process with the target city's data has better prediction performance than the studies done with a short training process with the target city's data. Secondly, weight transfer from Kocaeli to İstanbul reached more accurate test results than the weight transfer from Barcelona to İstanbul. Thirdly, transferring weights between Barcelona and Kocaeli did not show any improvement of performance to predict PM_{10} concentration by Method-2 and Method-3.

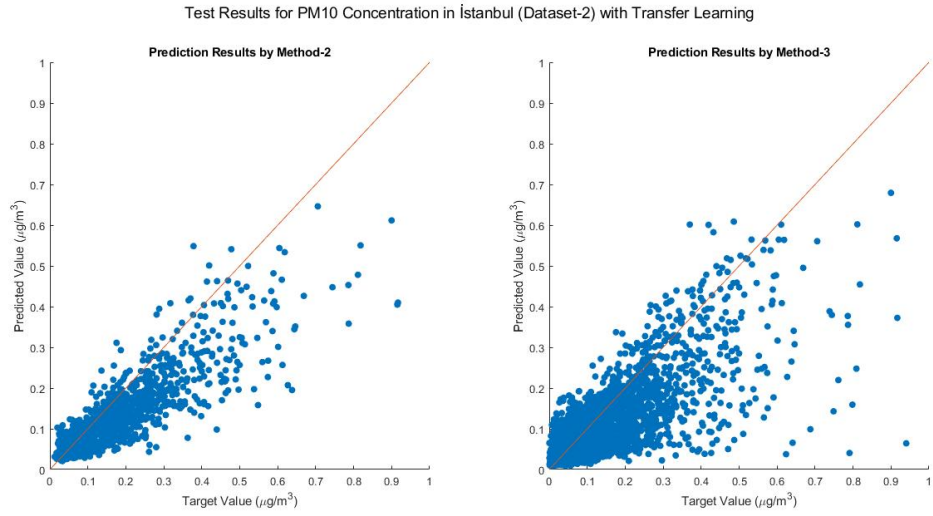


Figure 5.13: Scatter plot for PM_{10} concentration in İstanbul by Method-2 with transfer learning, pre-trained neural network with Kocaeli data are tested on İstanbul data.

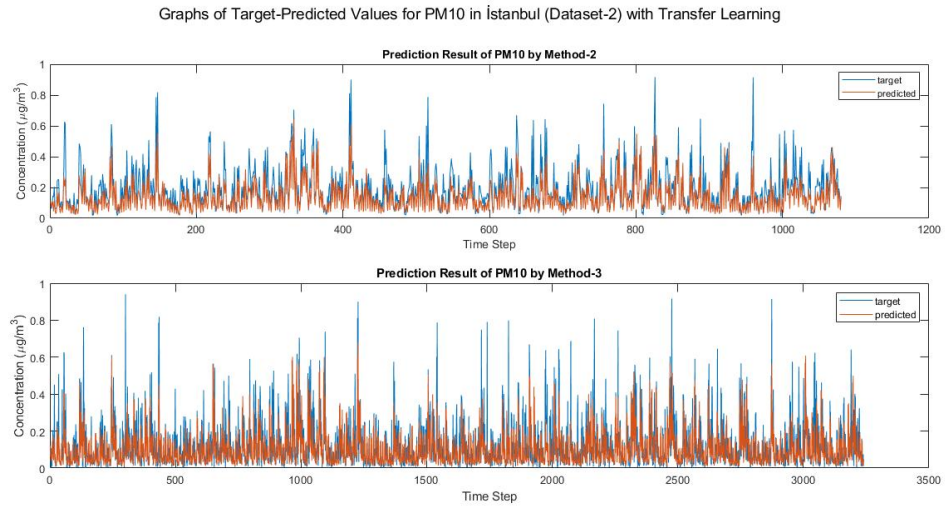


Figure 5.14: Scatter plot for PM_{10} concentration in İstanbul by Method-3 with transfer learning, pre-trained neural network with Kocaeli data are tested on İstanbul data.

6. CONCLUSION

In this study, a CNN+LSTM deep neural network model was developed to predict the future concentrations of air pollutants in different cities based on spatial and temporal features.

Air pollution is one of the most important reasons for early death diseases such as heart and lung diseases, stroke, lung cancer, asthma. Some air pollutants like O_3 and Nitrogen Oxides (NO_X) affect ecosystems and vegetation directly, get damage to water and soil that support the ecosystem. Also, air pollution has socio-economic effects like a decrease in the lifetime of individuals, an increase in medical cost, reducing productivity.

In recent years, the rapid development of IoT and smart cities make accurate environmental monitoring possible with updates on air pollution in real-time. Artificial intelligence-based air quality prediction models for smart cities can process data that comes from several IoT sensors placed over a large area and use machine learning algorithms to learn the correlation between features and make predictions accurately. The design of a reliable model needs to consider both spatial and temporal features to detect the regional historical trend of air pollution and the interaction of pollutants between regions.

Table 6.1 explains three different methods used in this study according to the input and output. Method-1 uses one pollutant's historical information to predict the concentration of that pollutant in the next hour. Method-2 uses historical information of three pollutants as input and outputs the concentration of one target pollutant in the next hour. Then, meteorological features are added to observe the

contribution of meteorological conditions to the prediction accuracy. In Method-3, the input contains the historical information of three pollutants, and the output is the concentration of three pollutants in the next hour, and the change in results is observed by adding meteorological data to the input. All three methods give the output as the concentration of the target pollutant/pollutants in all locations at the same time.

Table 6.1: Explanation of the methods used in the implementation

METHOD	INPUT STRUCTURE	INPUT	OUTPUT
Method-1	2-dimensional array	Historical information of one pollutant (univariate input)	Predicted concentration of one pollutant in the next hour (univariate output)
Method-2	3-dimensional array	Historical information of three pollutants (multivariate input)	Predicted concentration of one pollutant in the next hour (univariate output)
Method-3	3-dimensional array	Historical information of three pollutants (multivariate input)	Predicted concentration of three pollutants in the next hour (multivariate output)

First of all, Barcelona, Kocaeli, and İstanbul were chosen as target cities of the study. Air quality data included in the open-source of Barcelona’s City Hall Open Data Service contains hourly air pollution information between June 13, 2018, and January 31, 2019. In this data, the concentrations of the three pollutants are given in $\mu g/m^3$. The data for Kocaeli and İstanbul are released as an open-source data bank of the Republic of Turkey Ministry of Environment and Urban. The hourly data contain information in between 11/14/2017 17.00 - 4/11/2020 23.00 for Kocaeli and in between 01/01/2015 00.00 - 04/11/2020 23.00 for İstanbul. There is air quality information (concentration of pollutants in $\mu g/m^3$) and meteorological data (temperature, relative humidity, air pressure, wind speed, and wind direction).

Especially in Barcelona, the distances between locations (Eixample, Palau Reial, Vall Hebron) are not so far as to change the meteorological information between locations, therefore, only air pollution data were input to the deep learning model. In addition, the distance between selected three locations in the other two cities (Körfez,

Gebze and Alikahya for Kocaeli; Silivri, Esenyurt, and Sultangazi for İstanbul) is so much that it can lead to changes in the values of meteorological features. Table 6.2 shows the hyperparameter values that are set to reach the optimal prediction performance in each city.

Table 6.2: Optimal hyperparameter values for the air pollution prediction model

Hyperparameter	Barcelona	Kocaeli	İstanbul
Data Separation (Training, Validation, Test)	80%-10%-10%	70%-15%-15%	70%-15%-15%
Sample Selection	random	random	random
Number of Convolutional Layer	2	2	2
Number of Convolutional Filter (for one layer)	5	10	10
Number of LSTM Layer	1	1	1
Number of LSTM Unit (for one layer)	75	100	125
Activation Function	Sigmoid	Sigmoid	Sigmoid
Mini-Batch Size	30	100	200
Learning Rate	0.005	0.005	0.005
Frame Size	[8, 15]	[8, 15]	[8, 15]

The first objective of the thesis is to develop a supervised model for the prediction of air pollution from real sensor data obtained in different locations and to set the parameters that give an optimal level of learning based on the best performance metrics values achieved for the largest number of pollutants and cities. The CNN+LSTM based neural network predicts the hourly concentration of the air pollutants, in certain environmental factors (such as air pollution and meteorological information). While convolutional layers extract the relationships between locations for spatial features, LSTM layers extract temporal information characteristics from time series data. The nonlinear relationship between multivariable time series and air pollutants were combined, and the effects of air pollution and meteorological

data on prediction performance were observed. Generally, although the model selection task-related hyperparameters do not change among different pollutants, the parameters related to the training process change among the pollutants and the cities. The data properties and methods that give the best prediction results for different cities are given in Table 6.7.

Using different activation functions at the end of the network provided different prediction performance for each function. Although the use of ReLU after the fully connected layer eliminates negative values and it is useful while working with non-normalized data, it also leads to exceeding the target value for each pollutant while working with normalized data. On the other hand, sigmoid limits the prediction value between 0 and 1 in the last layer and improves the prediction performance. However, at higher target values, it brings the result of sigmoid prediction to lower points than the target value and the difference between prediction-target becomes more clear as the target value increases. Considering the test RMSE values, the error for the results with sigmoid is less than the error observed with ReLU. This is because the sigmoid captures small target values and ReLU gives prediction results exceeding the target values. Additionally, when using sigmoid in each learning layer in the neural network, the prediction model has failed. Therefore, the optimal use of the activation functions is ReLU in each hidden layer and sigmoid in the fully connected layer.

Table 6.3: Explanation of the methods and the data used for different cities

Properties	Barcelona	Kocaeli	İstanbul
Air Pollutants	O_3, NO_2, PM_{10}	PM_{10}, SO_2, NO_X	$O_3, NO_2, PM_{10}, SO_2, NO_X$
Meteorological Data	NO	Temperature, Humidity, Pressure, Wind Speed and Direction	Temperature, Humidity, Pressure, Wind Speed and Direction
Method-1	YES	NO	NO
Method-2	YES	YES	YES
Method-3	YES	YES	YES

While O_3 , NO_2 and PM_{10} were selected as the pollutants in Barcelona, PM_{10} , SO_2 and NO_X were selected in Kocaeli. For İstanbul, there are two different time series data sets; Dataset-1 was created from the information of the same pollutants with

Barcelona, Dataset-2 was created from the information of the same pollutants with Kocaeli. The model can catch the trend in test data for PM_{10} concentration, but the difference between target and predicted values increases at extreme points. In other words, the higher the target value, the higher the prediction error. In Barcelona, although both two methods have the same test RMSE values for all pollutants (0.12 for O_3 , 0.11 for NO_2 , and 0.07 for PM_{10}), Method-2 has higher correlation coefficients than Method-3 (Table 6.4). The correlation coefficient between target and predicted concentration was measured as 0.87 for O_3 , 0.83 for NO_2 , and 0.95 for PM_{10} with Method-2; 0.82 for O_3 , 0.73 for NO_2 , and 0.94 for PM_{10} with Method-3 (Table 6.5). In Kocaeli, Method-2 reached the RMSE values of 0.08 for PM_{10} , 0.07 for SO_2 and 0.08 for NO_X ; Method-3 reached 0.07 for PM_{10} , 0.09 for SO_2 and 0.08 for NO_X . In İstanbul, Method-2 reached the RMSE values of 0.12 for O_3 , 0.11 for NO_2 , 0.09 for NO_X , 0.09 for SO_2 and 0.08 for PM_{10} ; Method-3 reached 0.11 for O_3 , 0.10 for NO_2 , 0.10 for NO_X , 0.10 for SO_2 and 0.08 for PM_{10} (Table 6.6).

Table 6.3 explains the methods and the data properties used in different cities. Test RMSE values for PM_{10} prediction in Barcelona and Kocaeli are generally the same. In Barcelona, the RMSE value for O_3 and NO_2 in Barcelona is 5% higher than the RMSE value for PM_{10} . Although there was a 0.01 difference between RMSE values for different pollutants in Kocaeli, the model could not catch the trend in the test set for NO_X and SO_2 and generally produced predicted values lower than the target values. In İstanbul, the RMSE value for Nitrogen-based pollutants (NO_X and NO_2) is 0.04 higher than the RMSE for PM_{10} . The highest RMSE value was observed for O_3 and measured as 0.11. In general, the model gave lower prediction values than the target values in the studies for İstanbul when Method-2 was used.

Since the İstanbul data set is larger than the Barcelona and Kocaeli data set, İstanbul has more training samples than the other two cities. Nevertheless, RMSE values for O_3 , NO_2 , and NO_X are approximately the same as for other cities, while RMSE values for PM_{10} and SO_2 are higher than for the other two cities.

Table 6.4: Performance metrics for prediction results of Barcelona data

Performance Metrics for Barcelona			
Pollutant	Method	Test RMSE	Correlation Coefficient
O_3	Method-2	0.12	0.87
	Method-3	0.12	0.82
NO_2	Method-2	0.11	0.83
	Method-3	0.11	0.73
PM_{10}	Method-2	0.07	0.95
	Method-3	0.07	0.94

Table 6.5: Performance metrics for prediction results of Kocaeli data

Performance Metrics for Kocaeli			
Pollutant	Method	Test RMSE	Correlation Coefficient
PM_{10}	Method-2	0.08	0.85
	Method-3	0.07	0.86
SO_2	Method-2	0.07	0.74
	Method-3	0.09	0.74
NO_X	Method-2	0.08	0.77
	Method-3	0.09	0.79

Table 6.6: Performance metrics for prediction results of İstanbul data

Performance Metrics for İstanbul			
Pollutant	Method	Test RMSE	Correlation Coefficient
O_3	Method-2	0.12	0.89
	Method-3	0.11	0.86
NO_2	Method-2	0.11	0.82
	Method-3	0.10	0.69
PM_{10}	Method-2	0.08	0.75
	Method-3	0.08	0.70
SO_2	Method-2	0.09	0.79
	Method-3	0.10	0.68
NO_X	Method-2	0.09	0.86
	Method-3	0.10	0.78

Samples were created by using the 3D input structure to be used for Method-2 and Method-3 by adding meteorological features (temperature, relative humidity, air pressure, wind speed, and wind direction) to the historical air pollution data. As a result, adding meteorological information to the input for both Kocaeli and İstanbul generally increased the RMSE value by 10% to 40%. In other words, using only pollutants' concentrations as features makes the model more successful. When looking at the results in İstanbul, the use of meteorological data improves the prediction performance for PM_{10} concentration with both two methods with an improvement rate of about 12%. On the other hand, it caused an increase between 30% to 40% for the prediction of NO_X concentration. In Kocaeli, the use of meteorological data with air quality data improves the prediction performance only for SO_2 concentration with Method-3 (with an improvement rate of 10%). It resulted in an increase in RMSE value of 12% to 33% for other pollutants.

Table 6.7: Data properties and methods that give the best prediction results

Properties	Barcelona	Kocaeli	İstanbul
Pollutant	PM_{10}	PM_{10}	PM_{10}
Test RMSE	0.07	0.07	0.07
Method	Method-2 & Method-3	Method-3	Method-2
Input Structure	3-dimensional array	3-dimensional array	3-dimensional array
Data Normalization	YES	YES	YES
Meteorological Data	-	NO	YES
Features	Historical hourly concentration of pollutants (O_3 , NO_2 , PM_{10})	Historical hourly concentration of pollutants (PM_{10} , SO_2 , NO_X)	Historical hourly concentration of pollutants (PM_{10} , SO_2 , NO_X) and meteorological conditions
Input Type	Multivariate	Multivariate	Multivariate
Output Type	Univariate & Multivariate	Multivariate	Univariate

When looking at the results of the ANFIS model proposed with [44], the RMSE values are calculated as 0.25 for SO_2 , 0.20 for O_3 , and 0.16 for NO_2 . The algorithm proposed in this thesis reached a lower test RMSE for these pollutants in all three cities (Tables 6.4, 6.5, and 6.6) than the study in [44]. Method-2 and Method-3 is done with multivariate input, which is called "multivariate prediction" in [47]. Multivariate prediction results observed by Method-2 and Method-3 were compared with other multivariate prediction studies in the literature [29, 47]. For this purpose, the RMSE values reached by those studies were divided by the range of target pollutants in the cities where the studies were conducted. Thus, the RMSE values were converted into the same scale with the RMSE values in which this thesis was evaluated. In [29], the ANN prediction model is proposed to compare the performances of univariate and multivariate prediction. The RMSE value was calculated as 0.47 for O_3 and 0.11 for SO_2 by the prediction with univariate input, 0.21 for O_3 , and 0.15 for SO_2 by the prediction with multivariate input. Our CNN+LSTM model reached lower RMSE for O_3 in Barcelona and İstanbul, and SO_2 Kocaeli and İstanbul than the study [29]. In the second study done with multivariate input, they proposed an LSTM model to predict the PM_{10} concentration in time series data [47]. The RMSE value was calculated as 0.08 for prediction performance with both univariate and multivariate input. Looking at the results of our model with

multivariate input, the RMSE value was measured as 0.07 in Barcelona and Kocaeli, while it was 0.08 in İstanbul.

Table 6.8: Comparison of the prediction performance between 1-hidden layer LSTM model and CNN+LSTM model in Barcelona

Test RMSE Results in Barcelona				
Pollutant	Method	LSTM	CNN+LSTM	Improvement Rate
O_3	Method-2	RMSE=0.16	RMSE=0.12	25%
	Method-3	RMSE=0.15	RMSE=0.12	20%
NO_2	Method-2	RMSE=0.13	RMSE=0.11	15%
	Method-3	RMSE=0.12	RMSE=0.11	8%
PM_{10}	Method-2	RMSE=0.11	RMSE=0.07	36%
	Method-3	RMSE=0.14	RMSE=0.07	50%

Table 6.9: Comparison of the prediction performance between 1-hidden layer LSTM model and CNN+LSTM model in Kocaeli

Test RMSE Results in Kocaeli				
Pollutant	Method	LSTM	CNN+LSTM	Improvement Rate
PM_{10}	Method-2	RMSE=0.12	RMSE=0.07	42%
	Method-3	RMSE=0.17	RMSE=0.08	53%
SO_2	Method-2	RMSE=0.12	RMSE=0.09	25%
	Method-3	RMSE=0.13	RMSE=0.07	46%
NO_x	Method-2	RMSE=0.12	RMSE=0.09	25%
	Method-3	RMSE=0.13	RMSE=0.08	39%

To compare the prediction performance of the proposed CNN+LSTM deep neural network, a 1-hidden layer LSTM network is used to predict the concentration of the pollutants. The LSTM network was built with one input layer, one hidden LSTM layer of 75 units (neurons), a fully connected layer, and an output layer. Data sets used in the CNN+LSTM model were used to train and test the LSTM network, then test RMSE results were measured. When the RMSE results of the LSTM network and the CNN+LSTM deep neural network are compared, it is seen that the CNN+LSTM model is more successful. By modeling the inter-regional

Table 6.10: Comparison of the prediction performance between 1-hidden layer LSTM model and CNN+LSTM model in İstanbul

Test RMSE Results in İstanbul				
Pollutant	Method	LSTM	CNN+LSTM	Improvement Rate
O_3	Method-2	RMSE=0.13	RMSE=0.11	31%
	Method-3	RMSE=0.15	RMSE=0.12	20%
NO_2	Method-2	RMSE=0.10	RMSE=0.10	0%
	Method-3	RMSE=0.10	RMSE=0.11	-10%
PM_{10}	Method-2	RMSE=0.09	RMSE=0.08	11%
	Method-3	RMSE=0.10	RMSE=0.08	20%
SO_2	Method-2	RMSE=0.10	RMSE=0.10	0%
	Method-3	RMSE=0.11	RMSE=0.09	18%
NO_X	Method-2	RMSE=0.11	RMSE=0.10	9%
	Method-3	RMSE=0.17	RMSE=0.09	47%

spatial relationship, CNN learns the spread of pollutants into the environment and the interaction between each other. Tables 6.8, 6.9, and 6.10 give the comparison of prediction performance between 1-hidden layer LSTM network and CNN+LSTM deep neural network in terms of the improvement rate. The CNN+LSTM model has improvement rates between 11% to 53% for the prediction of PM_{10} concentration. The test RMSE was dropped from 0.11 and 0.14 to 0.07 with both methods in Barcelona, from 0.12 to 0.07 with Method-2 and from 0.17 to 0.08 with Method-3 in Kocaeli, from 0.09 and 0.10 to 0.08 with both methods in İstanbul. For the prediction of O_3 concentration with Method-2, CNN+LSTM model decrease RMSE from 0.16 to 0.12 by the improvement rate of 25% in Barcelona, from 0.13 to 0.11 by the improvement rate of 31% in İstanbul. CNN+LSTM model decreased the test RMSE from 0.15 to 0.12 by the improvement rate of 20% for O_3 prediction with Method-3 in Barcelona and İstanbul. For the prediction of NO_2 concentration in Barcelona, the CNN+LSTM model dropped RMSE from 0.13 and 0.12 to 0.11 by the improvement rate of 15% and 8% respectively. As for SO_2 , CNN+LSTM decreased the RMSE from 0.12 to 0.09 by the improvement rate of 25% with Method-2 and

0.13 to 0.07 by the improvement rate of 46% with Method-3 in Kocaeli. On the other hand, it decreased the RMSE from 0.11 to 0.09 by the improvement rate of 18% with only Method-3 in İstanbul. Lastly, the CNN+LSTM model improved the prediction performance of 1-hidden layer LSTM by the rates between 9% to 47%. In Kocaeli, it has a 25% improvement rate by reducing the RMSE with Method-2 from 0.12 to 0.09 and 39% improvement rate by reducing the RMSE with Method-3 from 0.12 to 0.09. In İstanbul, the RMSE was dropped from 0.11 to 0.10 with Method-2 by the improvement rate of 9% and 0.17 to 0.09 with Method-3 by the improvement rate of 47%.

The second principle objective is to transfer a model between cities and examine which trained model has prediction ability on weight transfer of the target city and whether the parameters that give an optimal level of learning are the same for different cities.

There are three types of transfer learning, transductive transfer learning is in which source and target tasks are the same where the domains have a different marginal probability distribution. In this study, transductive transfer learning is proposed to support the CNN+LSTM deep learning model. There are two ways to apply transfer learning to the target domain. Either the model can test directly on the target city's test data set, or a short training process with the target city's training data set can be applied to the network by setting a small learning rate before testing. The training process with the target domain's data set didn't increase the prediction accuracy, the lowest RMSE was observed when the model is applied directly to the target domain.

Also, the best transfer learning success has been achieved in weight transfer for Method-2 and Method-3 from Kocaeli to İstanbul, transfers between Barcelona-İstanbul or Barcelona-Kocaeli caused higher test RMSE than the transfer between Kocaeli-İstanbul. The reason is that Kocaeli and İstanbul are neighbor cities and have similar air pollution and meteorological characteristics. On the other hand,

the characteristics of Barcelona are not related to the other two cities.

Generally, the model selection task-related hyperparameters with optimal prediction performance are the same for different cities. However, the training process-related parameters and their values change according to cities.

Although a common deep neural network structure was determined, the transfer of the model did not increase the prediction performance in the target city. For a prediction model with high accuracy, it would be better to train the model with the data of each city and to set training related hyperparameters.



APPENDIX A: Descriptive Statistics

Table A.1: Descriptive sStatistics for Barcelona Data (before data preprocessing)

Barcelona (11/06/11/2018 09:00:00 AM - 31/01/31/2019 11:00:00 PM)						
Location	Pollutant	Min	Max	Average	St. Dev.	Median
Ciudadella	$O_3(\mu g/m^3)$	1	190	28.84	35.92	32
	$NO_2(\mu g/m^3)$	1	131	20.58	36.28	34
	$PM_{10}(\mu g/m^3)$	-	-	-	-	-
Eixample	$O_3(\mu g/m^3)$	1	174	24.76	31.86	28
	$NO_2(\mu g/m^3)$	6	197	22.76	52.52	51
	$PM_{10}(\mu g/m^3)$	6	109	9.90	26.33	26
Gracia	$O_3(\mu g/m^3)$	1	167	27.28	39.18	37
	$NO_2(\mu g/m^3)$	1	168	23.96	45.07	42
	$PM_{10}(\mu g/m^3)$	1	85	9.41	23.85	23
Palau Reial	$O_3(\mu g/m^3)$	1	351	27.39	42.69	41
	$NO_2(\mu g/m^3)$	1	146	19.29	27.62	22
	$PM_{10}(\mu g/m^3)$	4	48	6.465745	17.77050	18
Poblenou	$O_3(\mu g/m^3)$	-	-	-	-	-
	$NO_2(\mu g/m^3)$	3	121	19.79515	37.47208	35
	$PM_{10}(\mu g/m^3)$	4	113	11.48295	24.743890	24
Sants	$O_3(\mu g/m^3)$	-	-	-	-	-
	$NO_2(\mu g/m^3)$	3	165	19.60329	32.543846	27
	$PM_{10}(\mu g/m^3)$	-	-	-	-	-
Vall Hebron	$O_3(\mu g/m^3)$	1	187	30.38234	51.56068	49
	$NO_2(\mu g/m^3)$	2	114	18.88299	28.698912	23
	$PM_{10}(\mu g/m^3)$	2	38	6.332675	19.06212	20

Table A.2: Descriptive sStatistics for Kocaeli Data (before data preprocessing)

Kocaeli MTHM (114/141/2017 17:00:56 PM - 4/4/11/2020 11:00:56 PM)						
Location	Pollutant	Min	Max	Average	St. Dev.	Median
Alikahya	$PM_{10}(\mu g/m^3)$	1.25	314.93	48.66	31.92	40.90
	$SO_2(\mu g/m^3)$	1.00	169.64	6.29	10.54	2.71
	$NO_X(\mu g/m^3)$	1.00	619.22	63.08	61.60	43.00
Gebze	$PM_{10}(\mu g/m^3)$	1.01	540.65	29.45	27.76	22.16
	$SO_2(\mu g/m^3)$	1.00	221.90	11.42	18.65	7.70
	$NO_X(\mu g/m^3)$	1.00	1950.68	111.28	163.11	56.40
Krfez	$PM_{10}(\mu g/m^3)$	4.91	604.40	46.15	27.01	39.84
	$SO_2(\mu g/m^3)$	1.00	312.91	7.77	14.24	3.24
	$NO_X(\mu g/m^3)$	1.20	609.96	63.55	62.11	44.08

Table A.3: Descriptive Statistics for İstanbul Data (before data preprocessing)

İstanbul MTHM (1/11/2015 12:00:56 AM - 4/4/11/2020 11:00:56 PM)						
Esenyurt	$PM_{10}(\mu g/m^3)$	1.00	985.00	76.97	73.07	56.13
	$SO_2(\mu g/m^3)$	1.00	220.26	6.56	6.49	4.57
	$NO_2(\mu g/m^3)$	0.02	152.26	25.14	17.44	20.61
	$NO_X(\mu g/m^3)$	0.08	1421.69	88.16	118.64	51.74
	$O_3(\mu g/m^3)$	1.00	924.91	33.46	25.91	29.70
Silivri	$PM_{10}(\mu g/m^3)$	1.00	782.49	34.32	27.28	27.90
	$SO_2(\mu g/m^3)$	0.01	83.61	3.28	4.23	1.91
	$NO_2(\mu g/m^3)$	0.64	272.27	24.34	19.31	17.39
	$NO_X(\mu g/m^3)$	0.24	1101.74	45.29	68.73	22.92
	$O_3(\mu g/m^3)$	0.01	783.92	56.19	31.93	58.99
Sultangazi	$PM_{10}(\mu g/m^3)$	1.37	454.46	59.04	39.81	51.03
	$SO_2(\mu g/m^3)$	0.01	86.41	4.08	4.68	2.93
	$NO_2(\mu g/m^3)$	0.01	175.34	34.38	20.87	30.28
	$NO_X(\mu g/m^3)$	0.05	1046.46	74.38	72.62	54.39
	$O_3(\mu g/m^3)$	1.00	710.22	35.46	24.42	34.29

APPENDIX B: Histogram for the Concentration of Pollutants in Cities

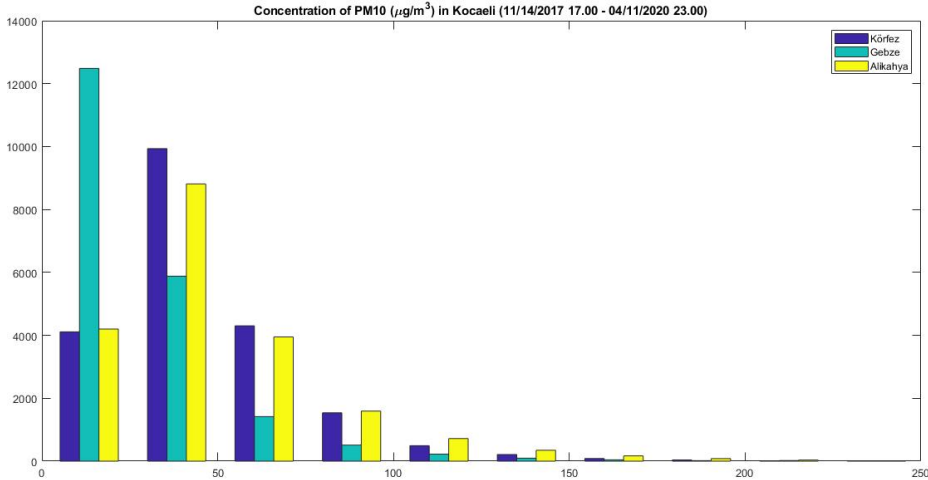


Figure B.1: Histogram for the concentration of PM_{10} in Kocaeli

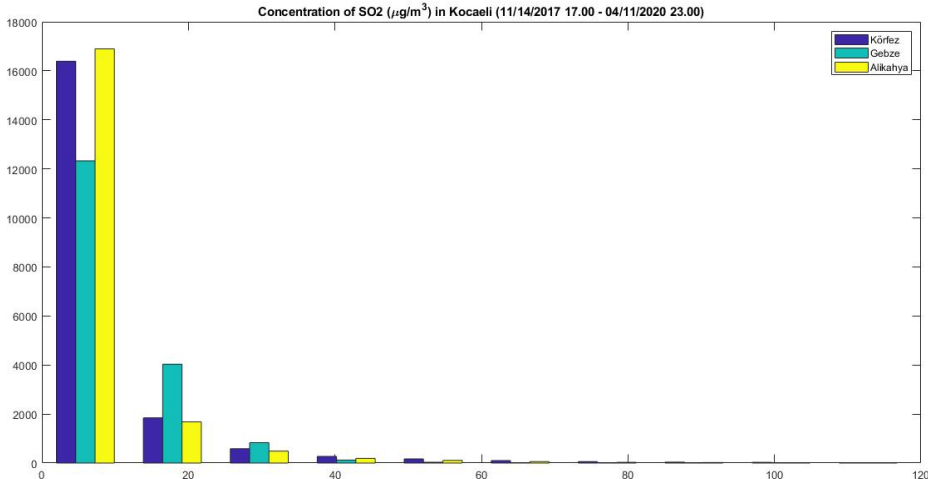


Figure B.2: Histogram for the concentration of SO_2 in Kocaeli

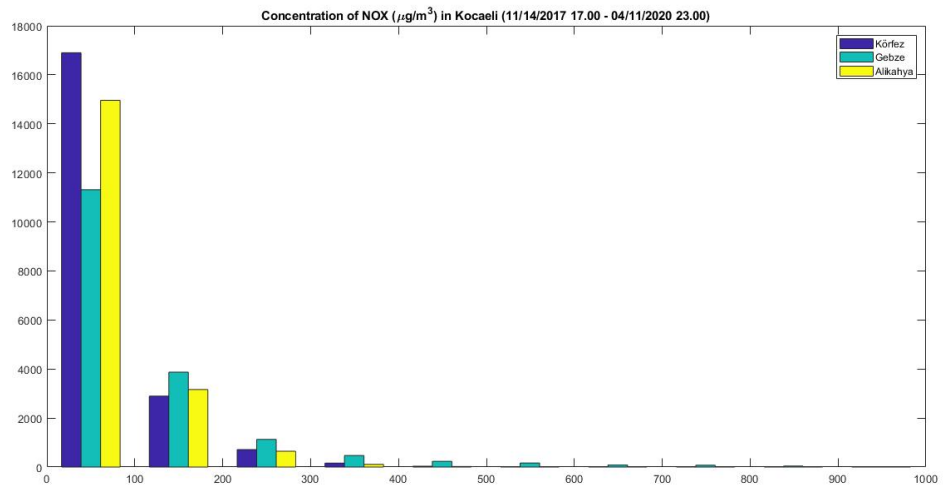


Figure B.3: Histogram for the concentration of NO_x in Kocaeli

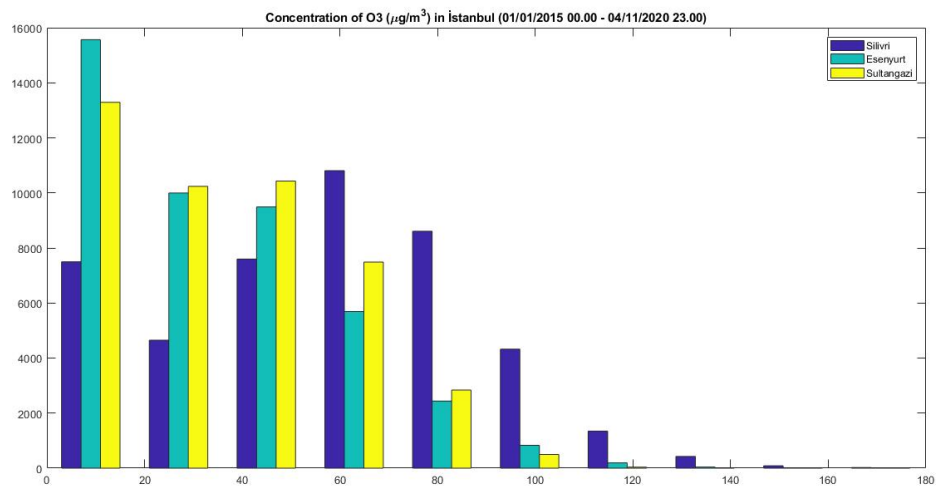


Figure B.4: Histogram for the concentration of O_3 in İstanbul

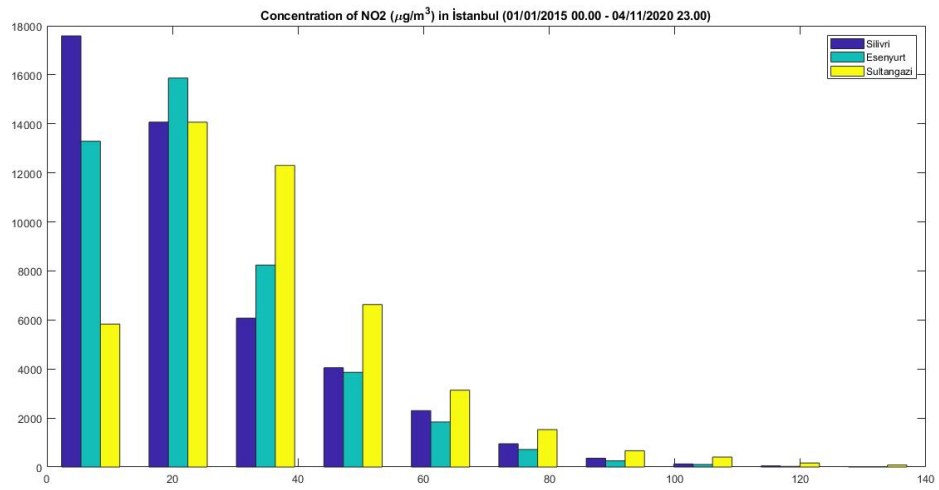


Figure B.5: Histogram for the concentration of NO_2 in İstanbul

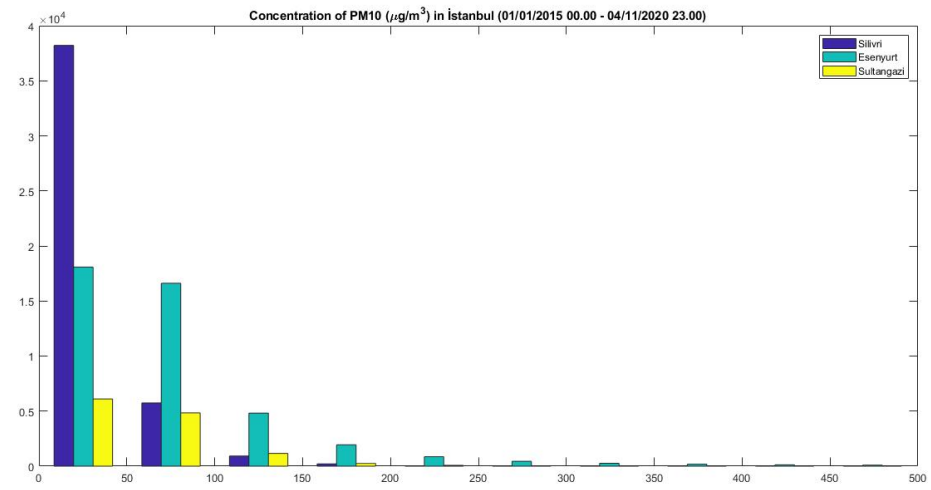


Figure B.6: Histogram for the concentration of PM_{10} in İstanbul

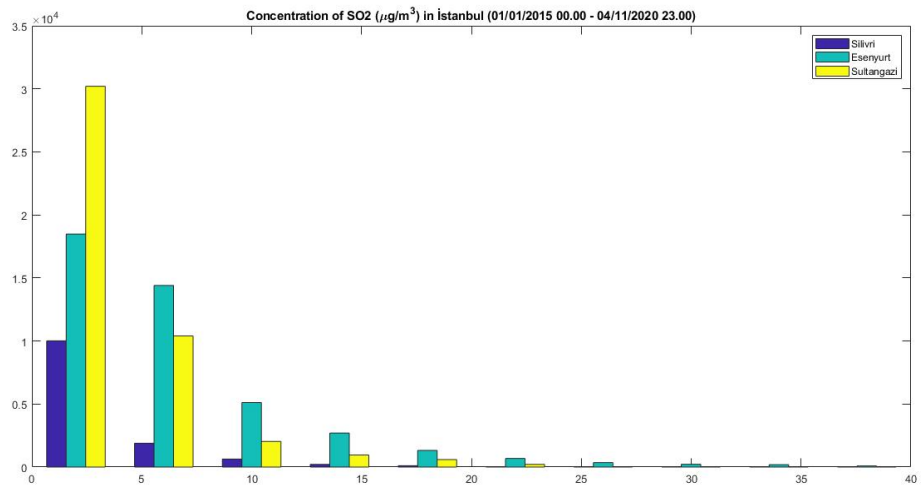


Figure B.7: Histogram for the concentration of SO_2 in İstanbul

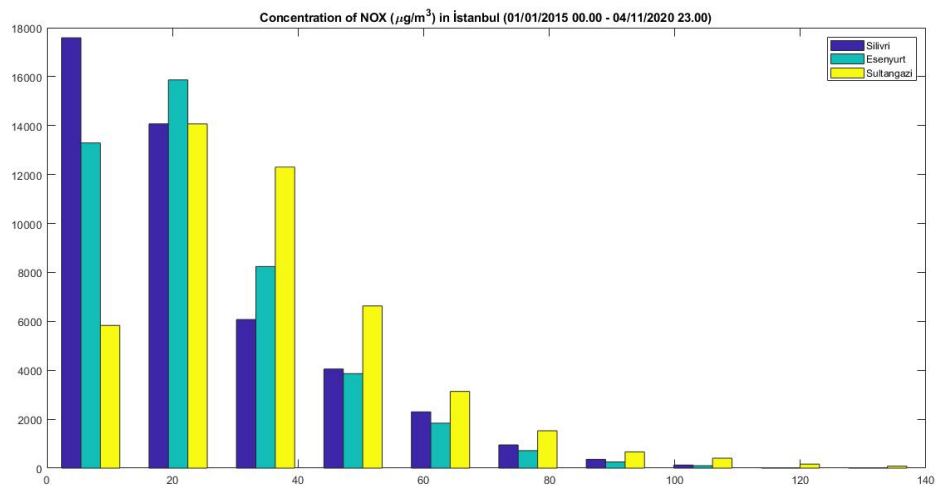


Figure B.8: Histogram for the concentration of NO_X in İstanbul

APPENDIX C: Time Series Graphs

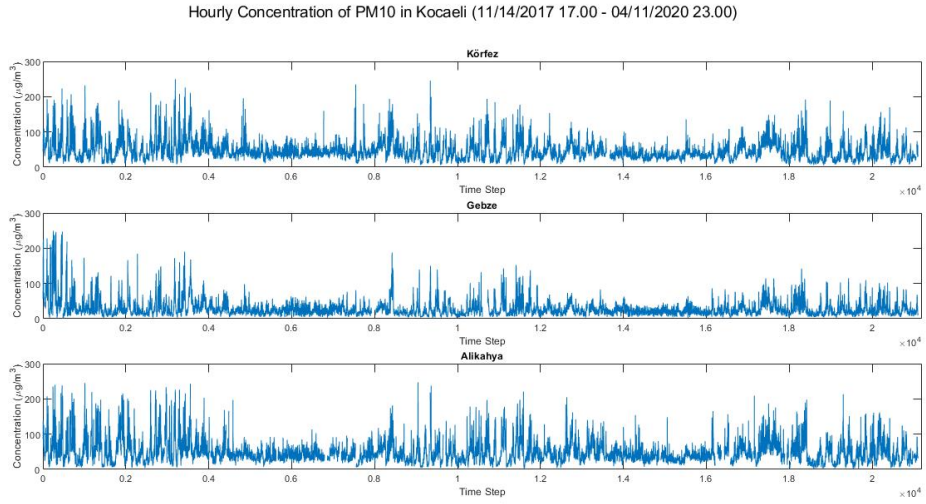


Figure C.1: Hourly concentration of PM_{10} in Kocaeli

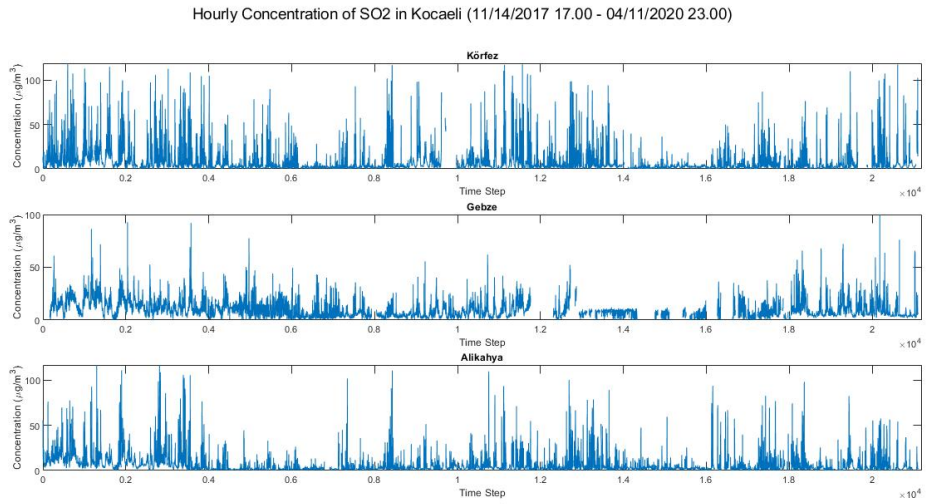


Figure C.2: Hourly concentration of SO_2 in Kocaeli

Hourly Concentration of NO_x in Kocaeli (11/14/2017 17.00 - 04/11/2020 23.00)

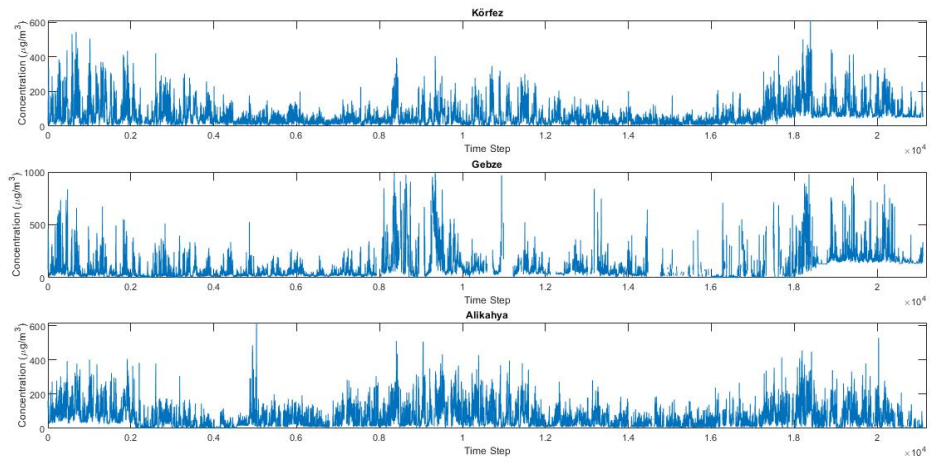


Figure C.3: Hourly concentration of NO_x in Kocaeli

Hourly Concentration of O₃ in İstanbul (01/01/2015 10.00 - 04/11/2020 23.00)

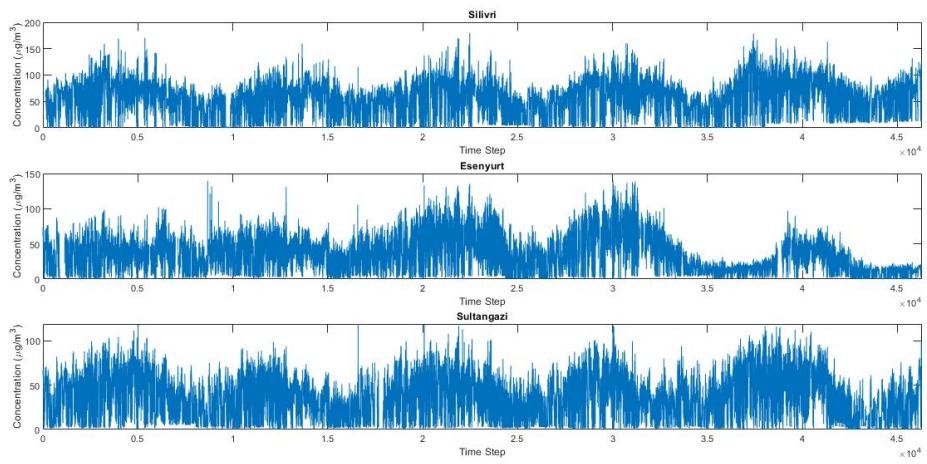


Figure C.4: Hourly concentration of O₃ in İstanbul

Hourly Concentration of NO₂ in İstanbul (01/01/2015 10.00 - 04/11/2020 23.00)

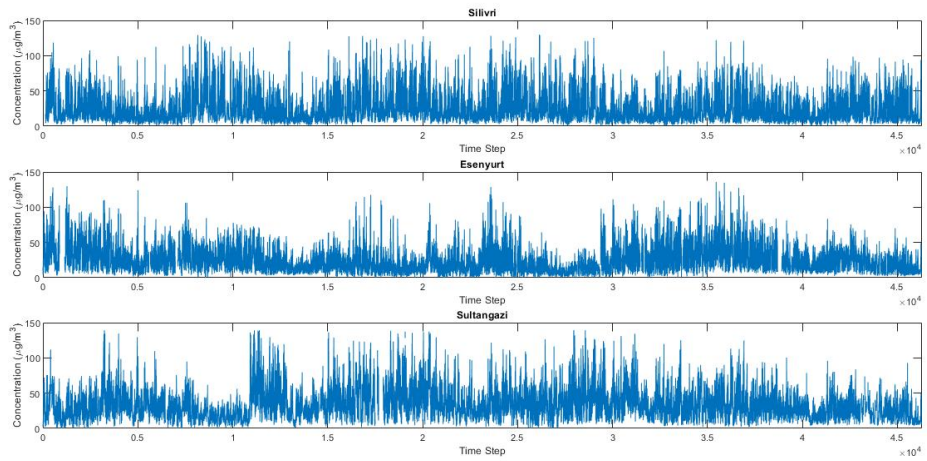


Figure C.5: Hourly concentration of NO_2 in İstanbul

Hourly Concentration of PM₁₀ in İstanbul (01/01/2015 10.00 - 04/11/2020 23.00)

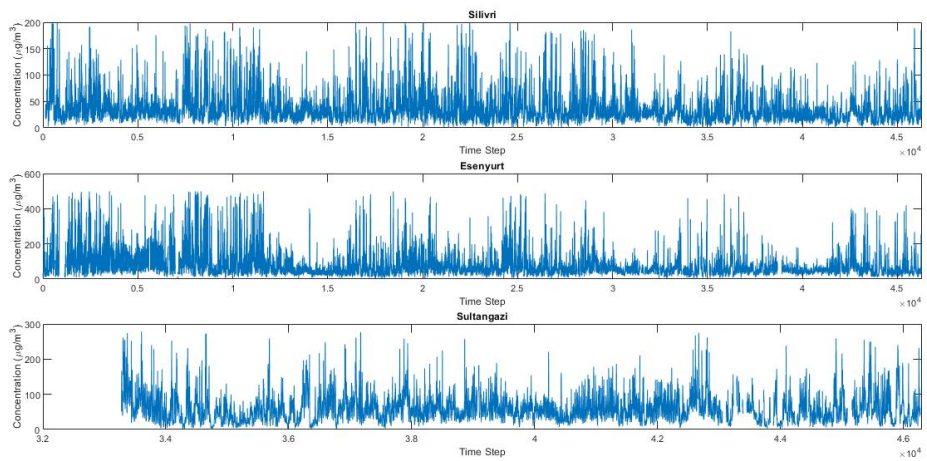


Figure C.6: Hourly concentration of PM_{10} in İstanbul

Hourly Concentration of SO₂ in İstanbul (01/01/2015 10.00 - 04/11/2020 23.00)

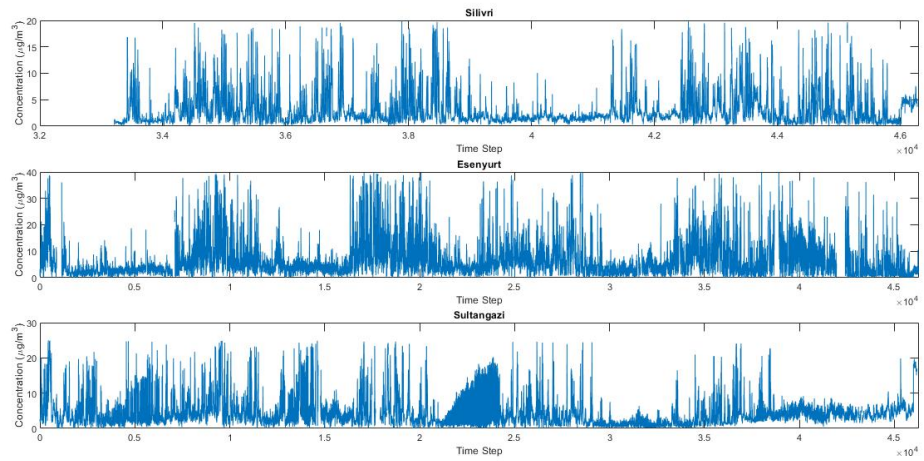


Figure C.7: Hourly concentration of SO₂ in İstanbul

Hourly Concentration of NO_x in İstanbul (01/01/2015 10.00 - 04/11/2020 23.00)

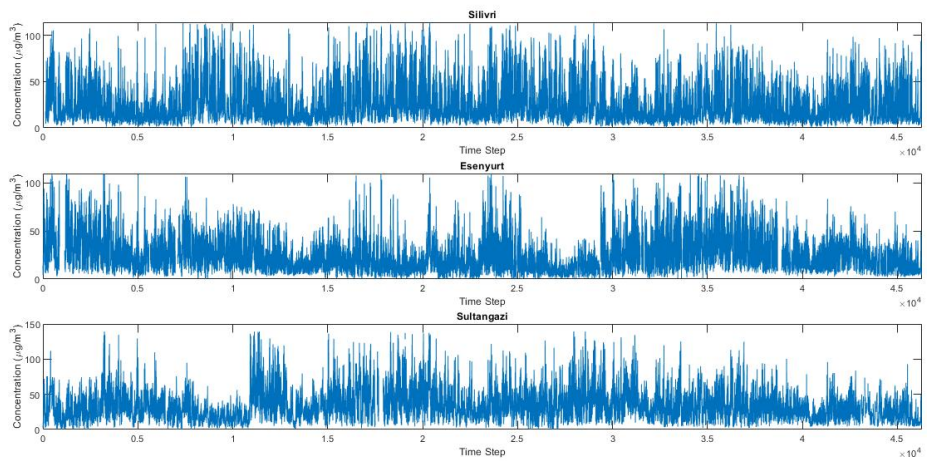


Figure C.8: Hourly concentration of NO_x in İstanbul

APPENDIX D: Results of Barcelona

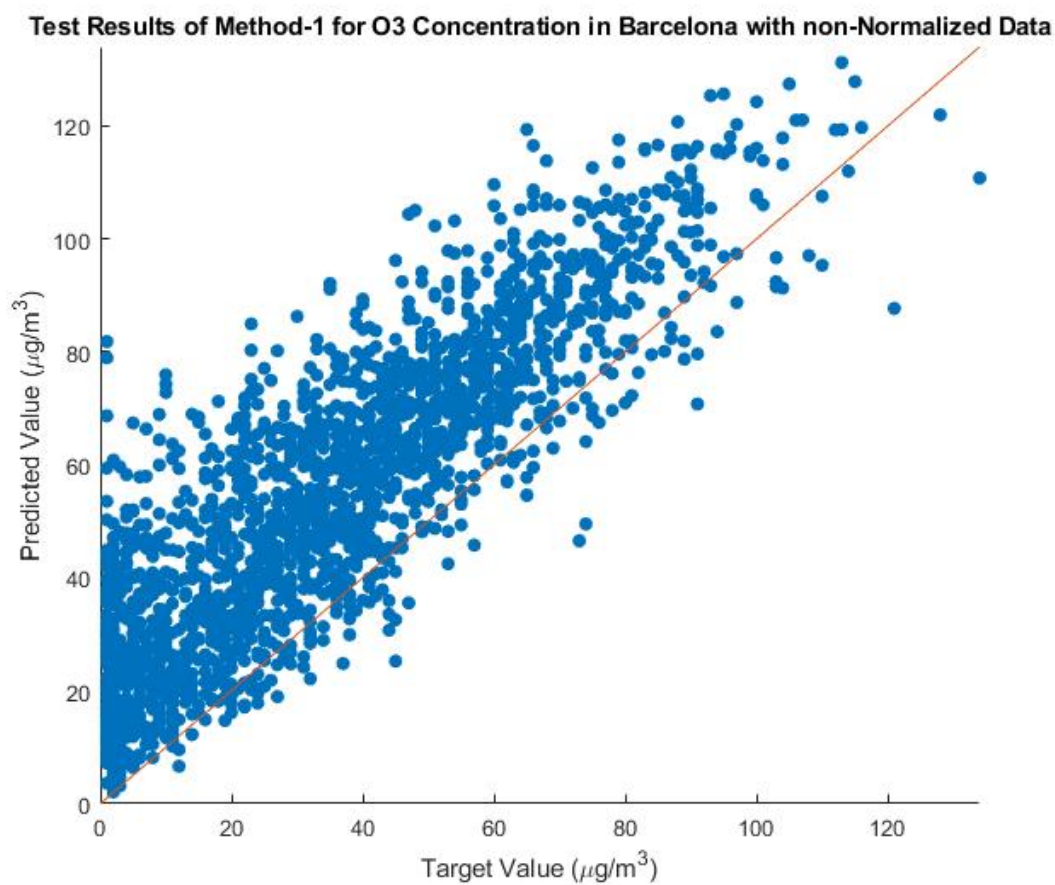


Figure D.1: Scatter plot of Method-1 test results for O_3 in Barcelona with non-normalized data

Test Results of Method-1 for NO₂ Concentration in Barcelona with non-Normalized Data

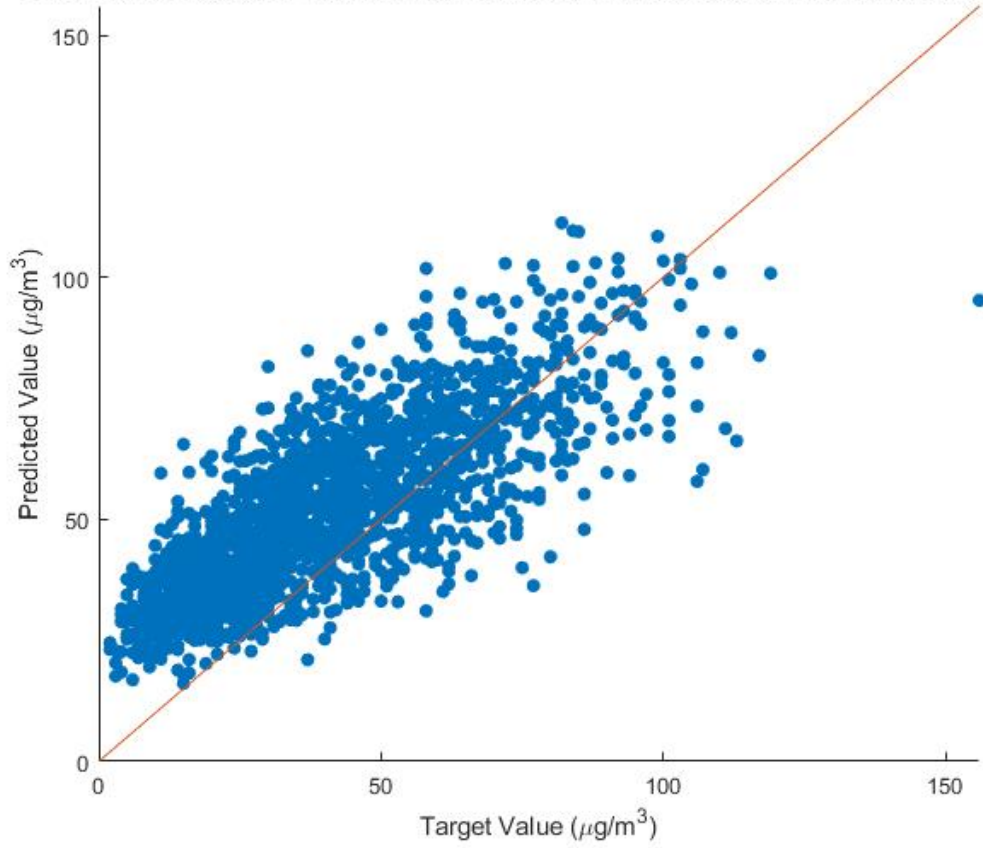


Figure D.2: Scatter plot of Method-1 test results for NO_2 in Barcelona with non-normalized data

Test Results of Method-1 for PM10 Concentration in Barcelona with non-Normalized Data

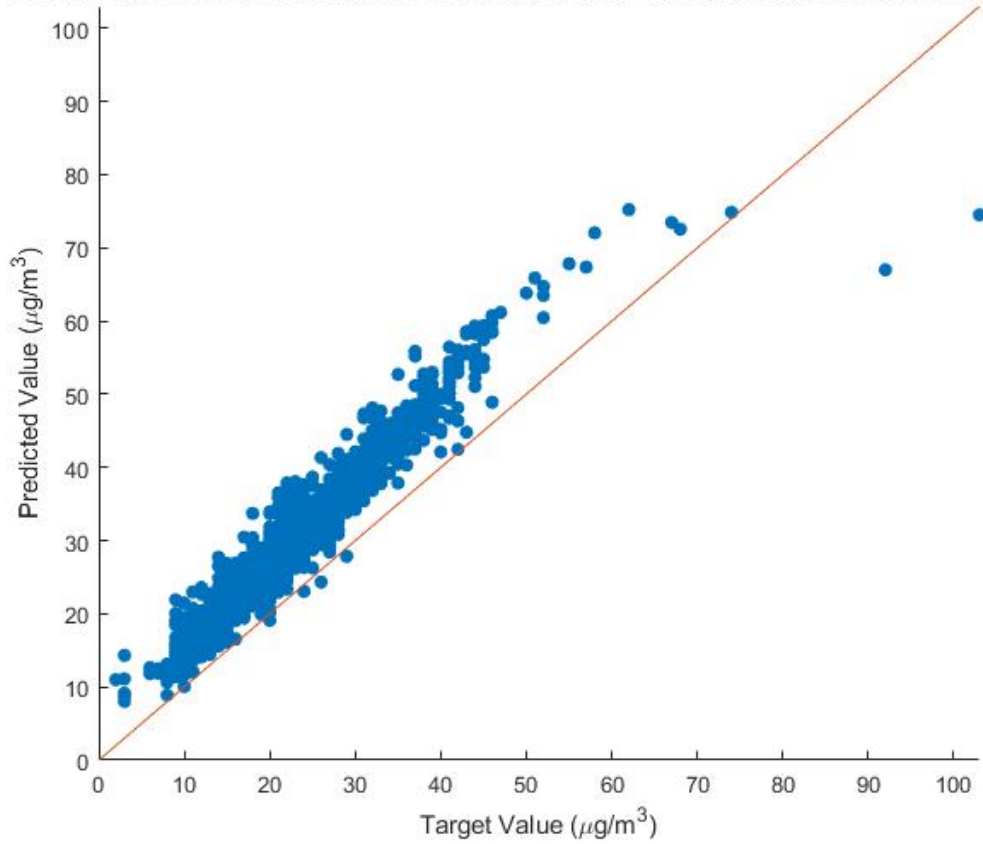


Figure D.3: Scatter plot of Method-1 test results for PM_{10} in Barcelona with non-normalized data

Target-Predicted Concentration of O3 in Barcelona with non-Normalized Data (by Method-1)

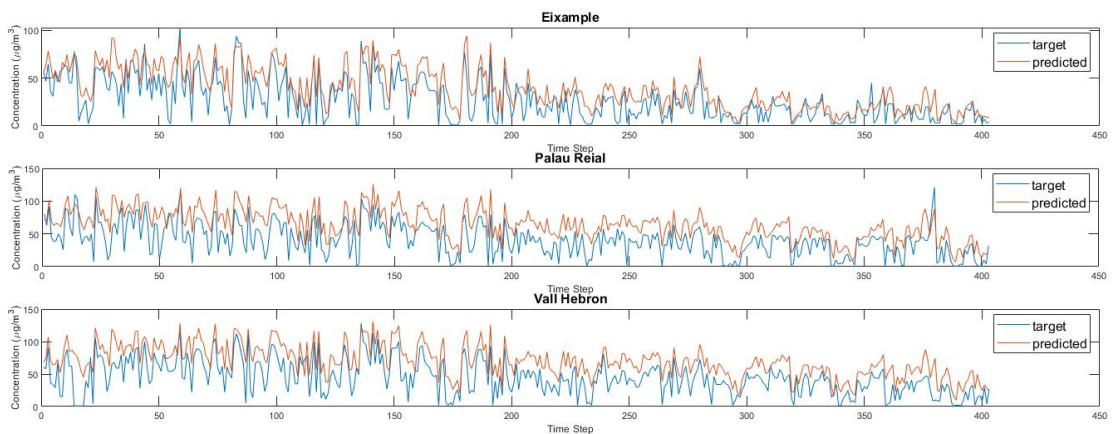


Figure D.4: Graph of the target-predicted values of Method-1 test results for O_3 in Barcelona with non-normalized data

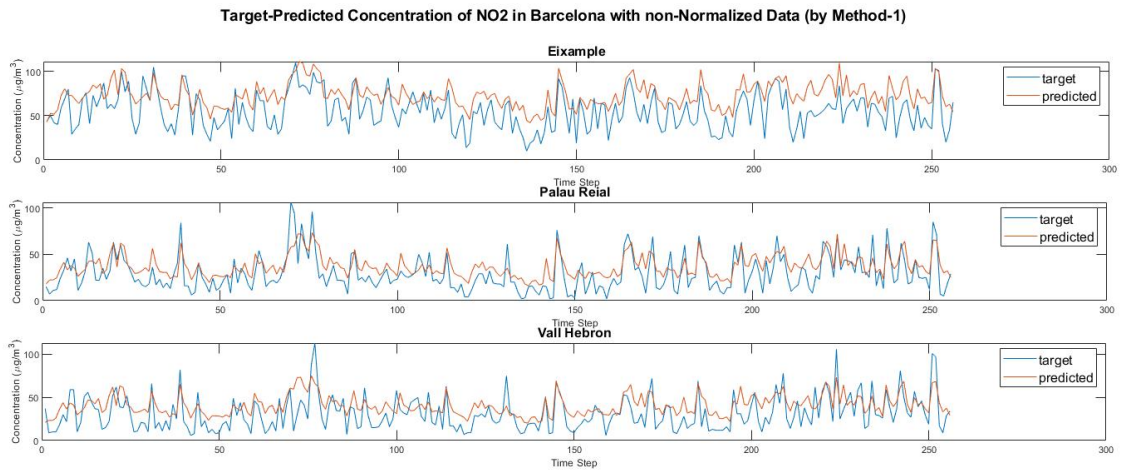


Figure D.5: Graph of the target-predicted values of Method-1 test results for NO_2 in Barcelona with non-normalized data

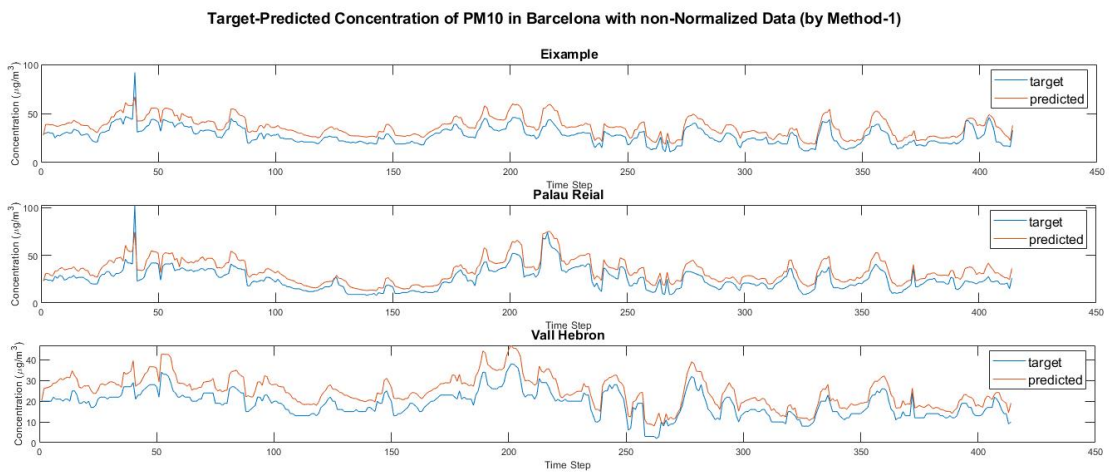


Figure D.6: Graph of the target-predicted values of Method-1 test results for PM_{10} in Barcelona with non-normalized data

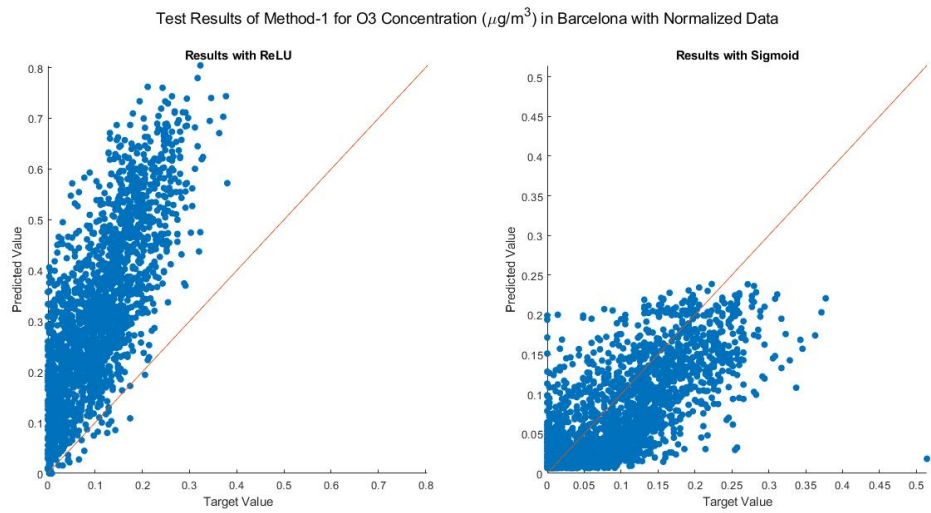


Figure D.7: Scatter plot of Method-1 test results for O_3 in Barcelona with normalized data

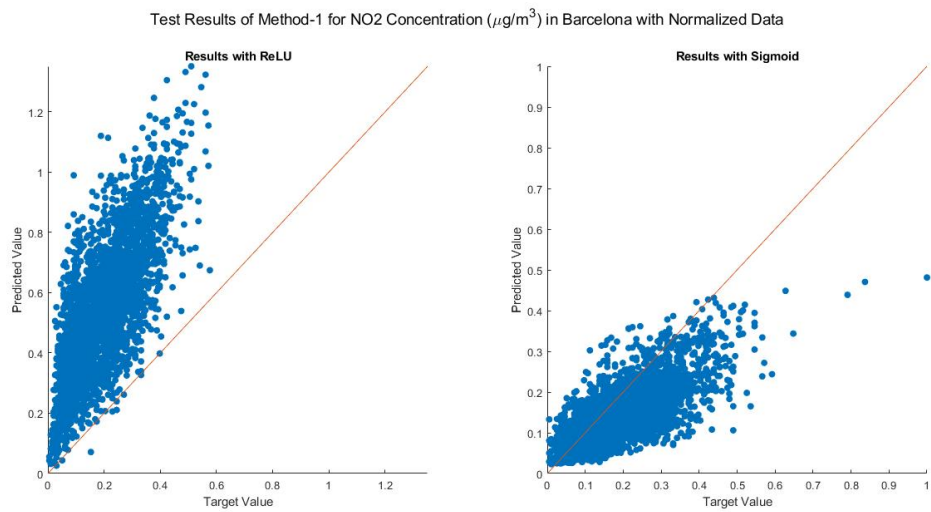


Figure D.8: Scatter plot of Method-1 test results for NO_2 in Barcelona with normalized data

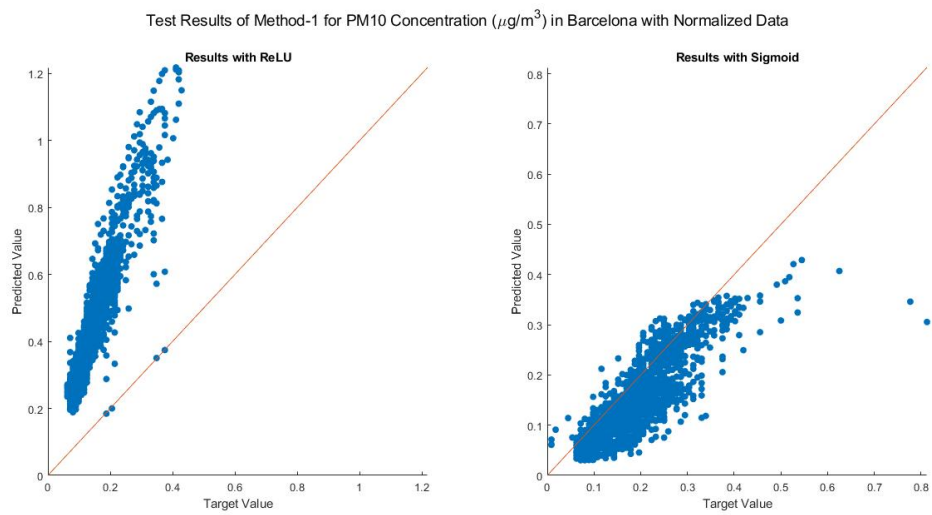


Figure D.9: Scatter plot of Method-1 test results for PM_{10} in Barcelona with normalized data

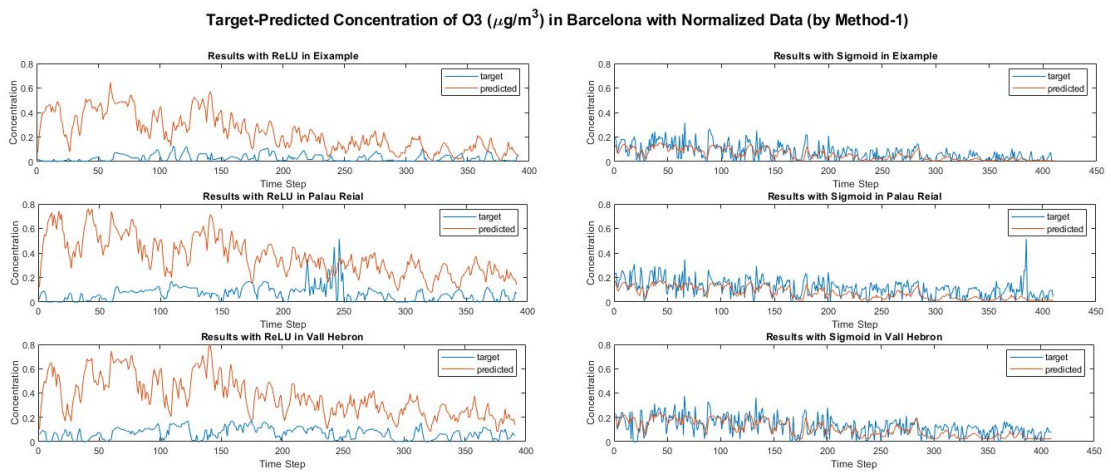


Figure D.10: Graphs of the target-predicted values of Method-1 test results for O_3 in Barcelona with normalized data

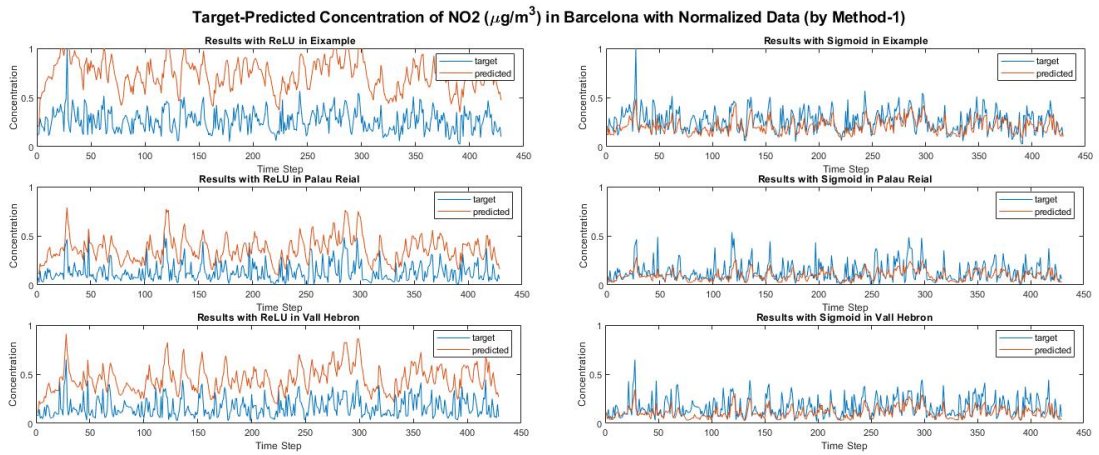


Figure D.11: Graphs of the target-predicted values of Method-1 test results for NO_2 in Barcelona with normalized data

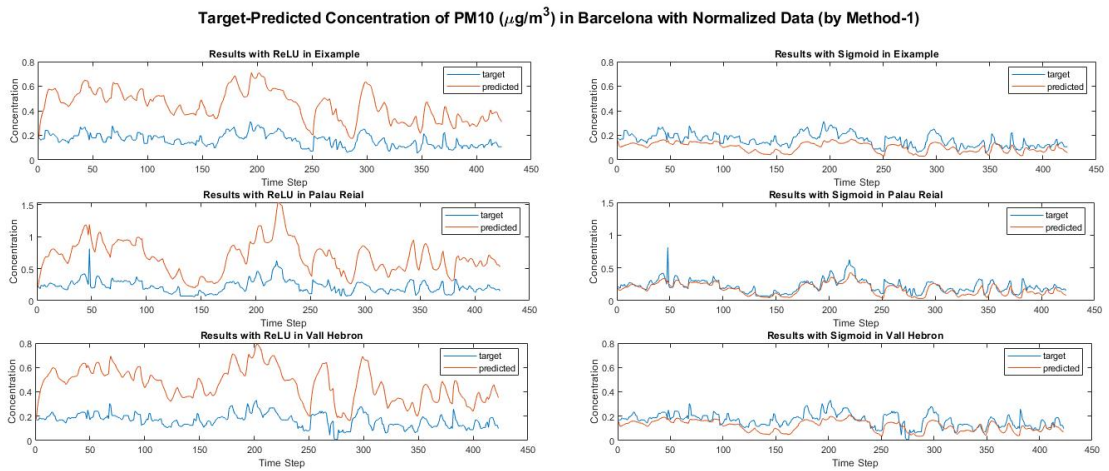


Figure D.12: Graphs of the target-predicted values of Method-1 test results for PM_{10} in Barcelona with normalized data

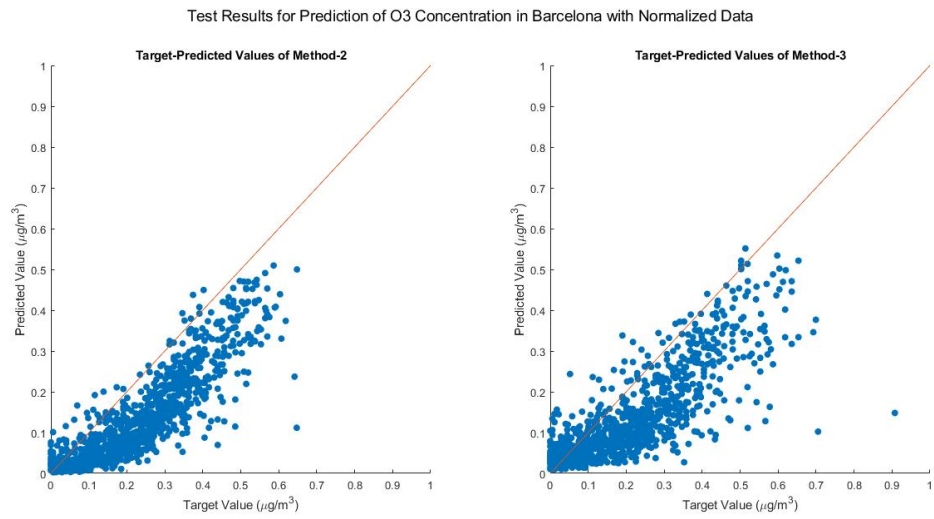


Figure D.13: Scatter plot of Method-2 and Method-3 test results for O_3 in Barcelona with normalized data

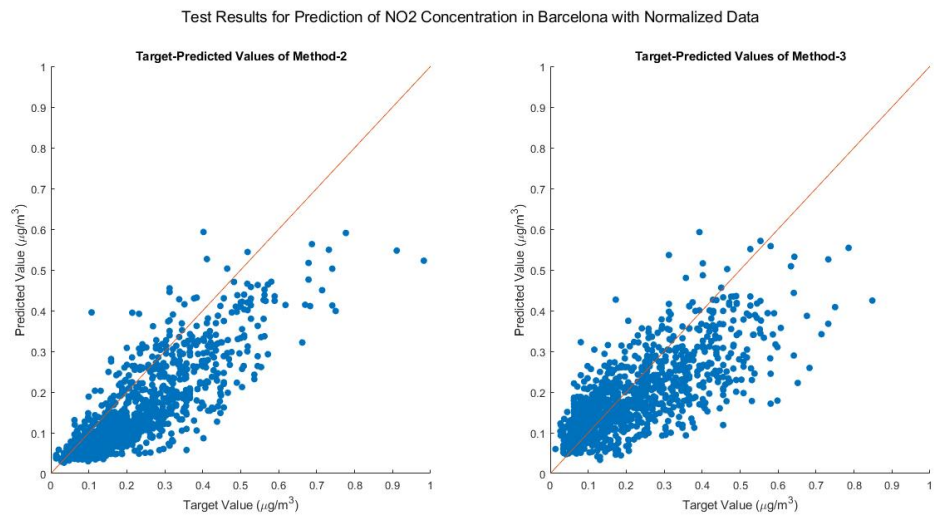


Figure D.14: Scatter plot of Method-2 and Method-3 test results for NO_2 in Barcelona with normalized data

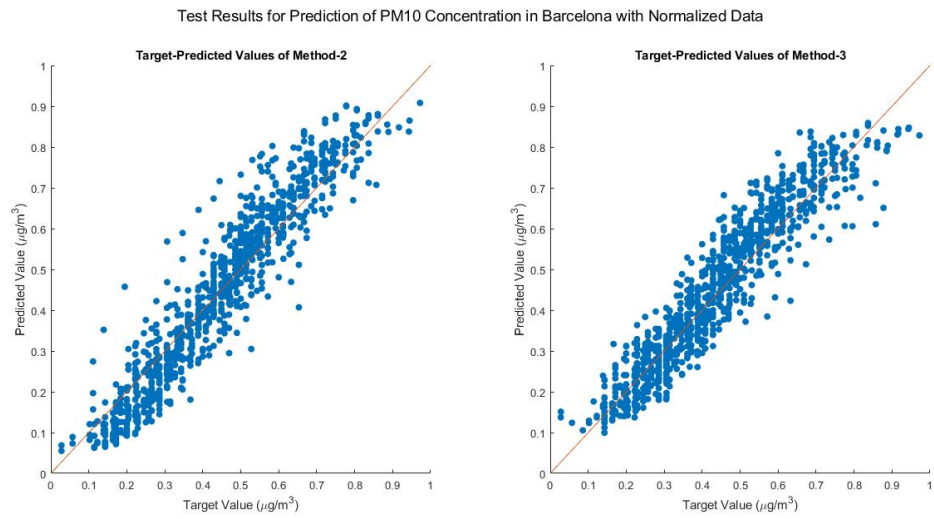


Figure D.15: Scatter plot of Method-2 and Method-3 test results for PM_{10} in Barcelona with normalized data

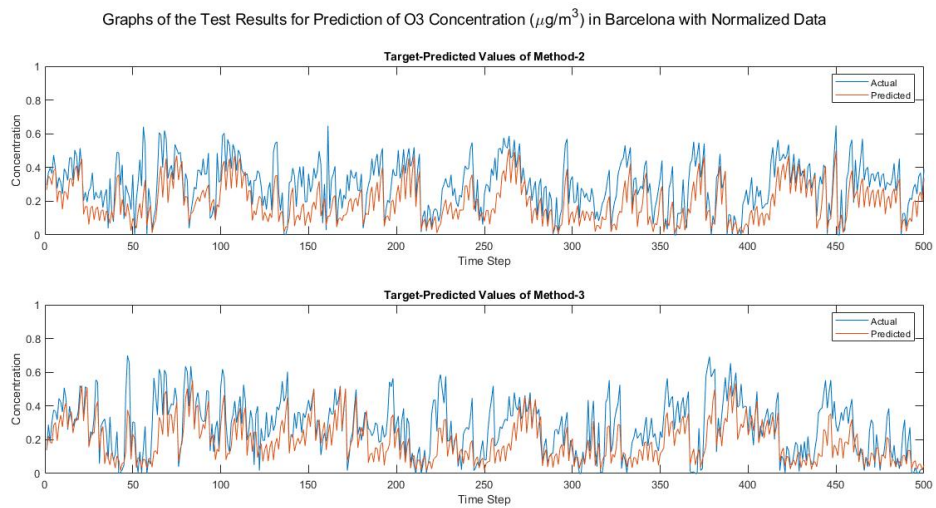


Figure D.16: Graphs of the target-predicted values of Method-2 and Method-3 test results for O_3 in Barcelona with normalized data

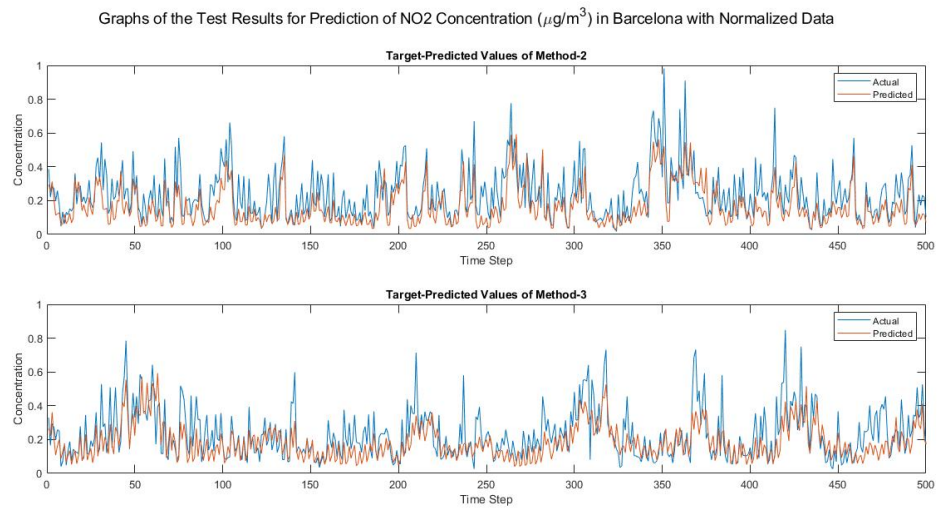


Figure D.17: Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_2 in Barcelona with normalized data

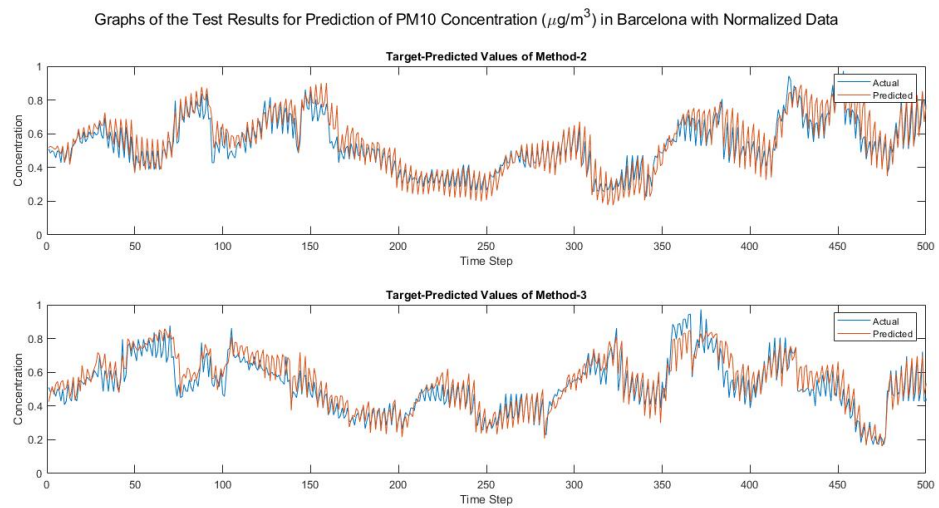


Figure D.18: Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in Barcelona with normalized data

APPENDIX E: Results of Kocaeli

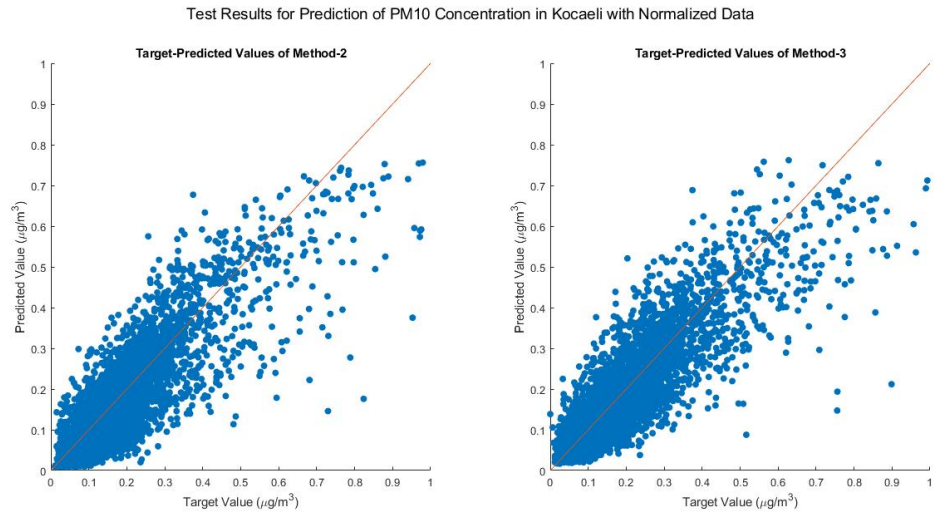


Figure E.1: Scatter plot of Method-2 and Method-3 test results for PM_{10} in Kocaeli with normalized data

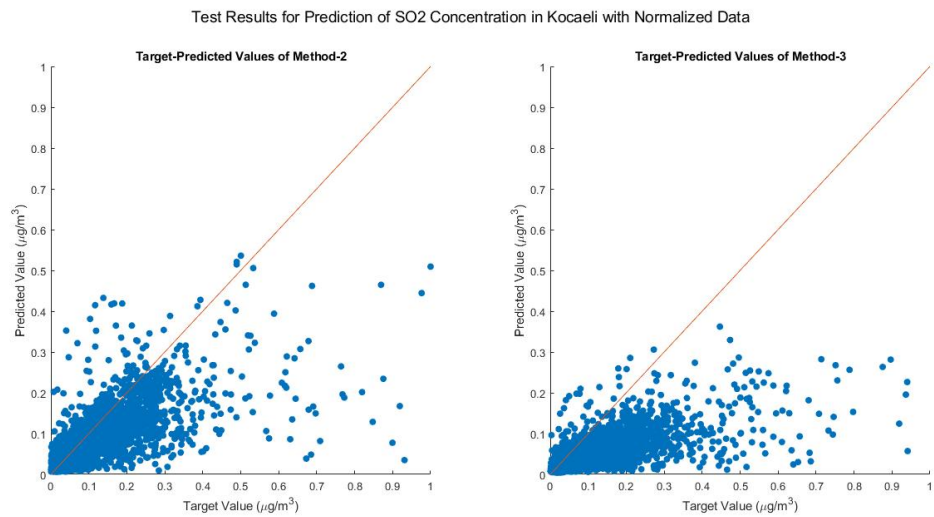


Figure E.2: Scatter plot of Method-2 and Method-3 test results for SO_2 in Kocaeli with normalized data

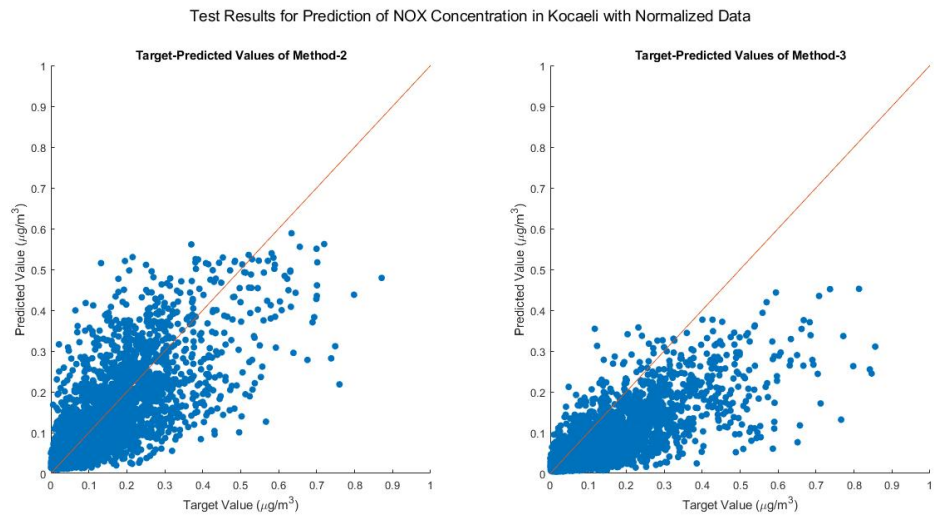


Figure E.3: Scatter plot of Method-2 and Method-3 test results for NO_X in Kocaeli with normalized data

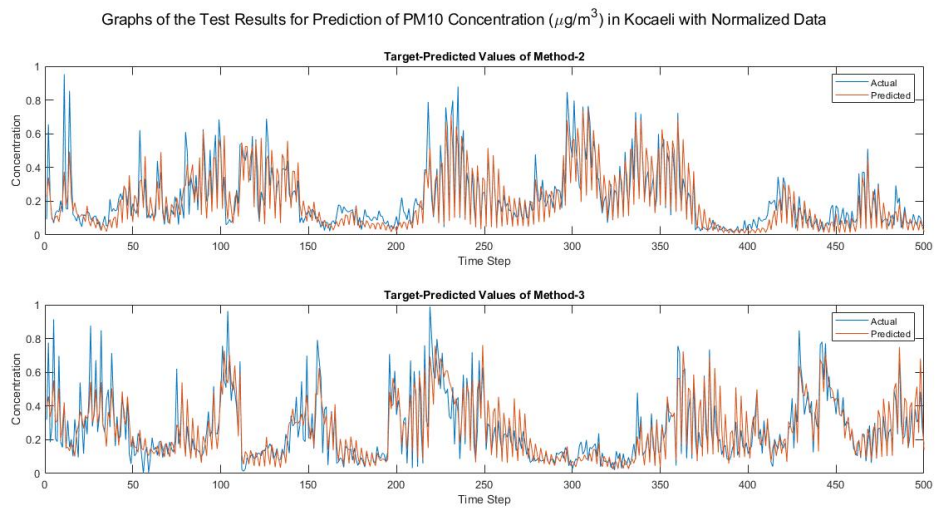


Figure E.4: Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in Kocaeli with normalized data

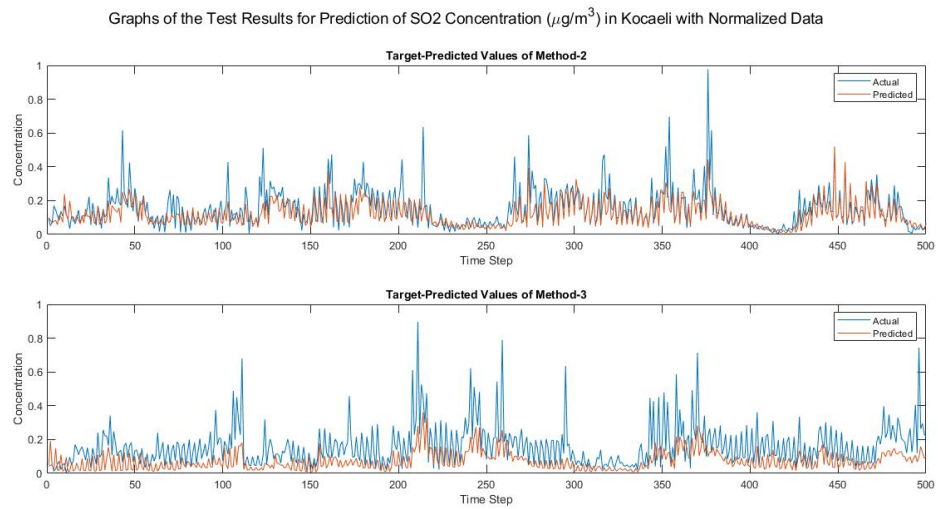


Figure E.5: Graphs of the target-predicted values of Method-2 and Method-3 test results for SO_2 in Kocaeli with normalized data

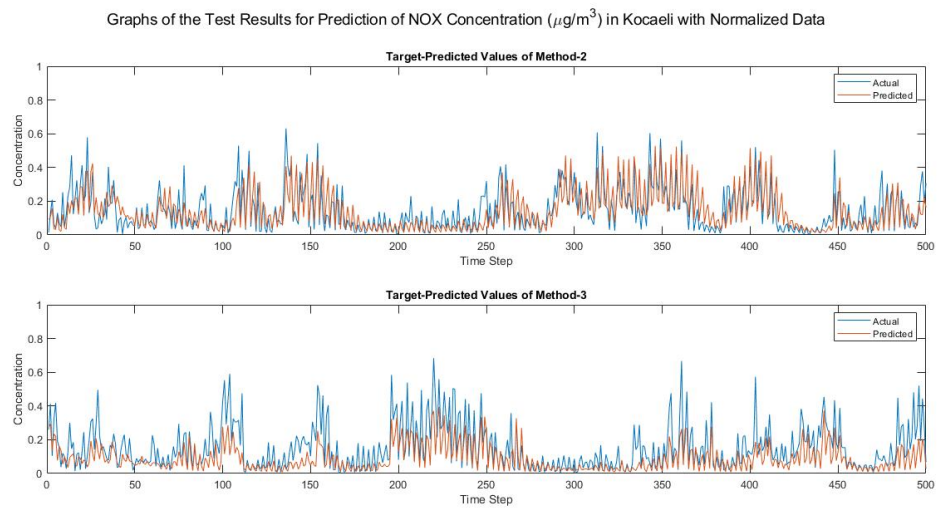


Figure E.6: Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_X in Kocaeli with normalized data

APPENDIX F: Results of İstanbul Dataset-1

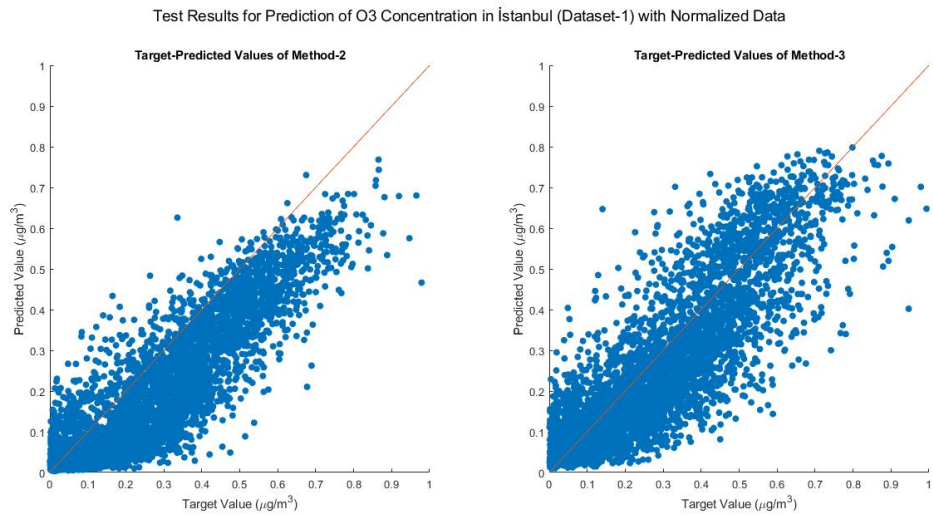


Figure F.1: Scatter plot of Method-2 and Method-3 test results for O_3 in İstanbul (Dataset-1) with normalized data

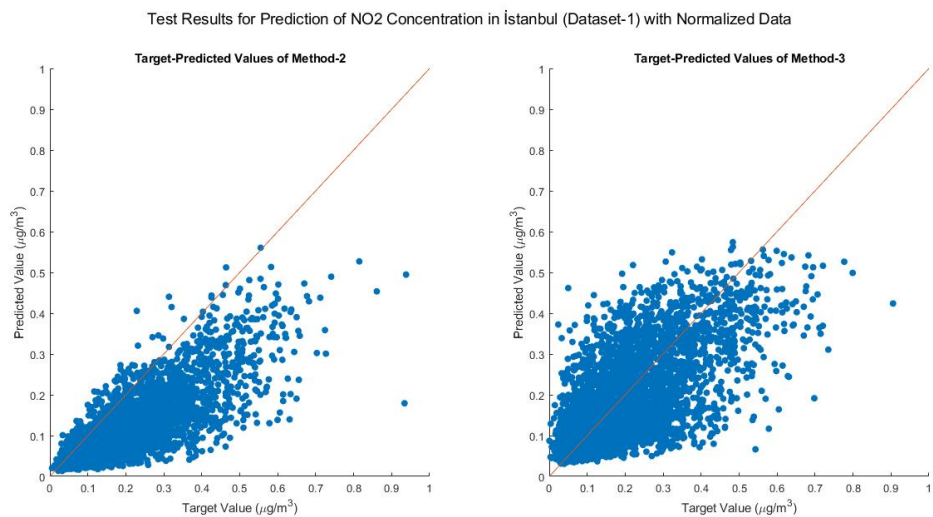


Figure F.2: Scatter plot of Method-2 and Method-3 test results for NO_2 in İstanbul (Dataset-1) with normalized data

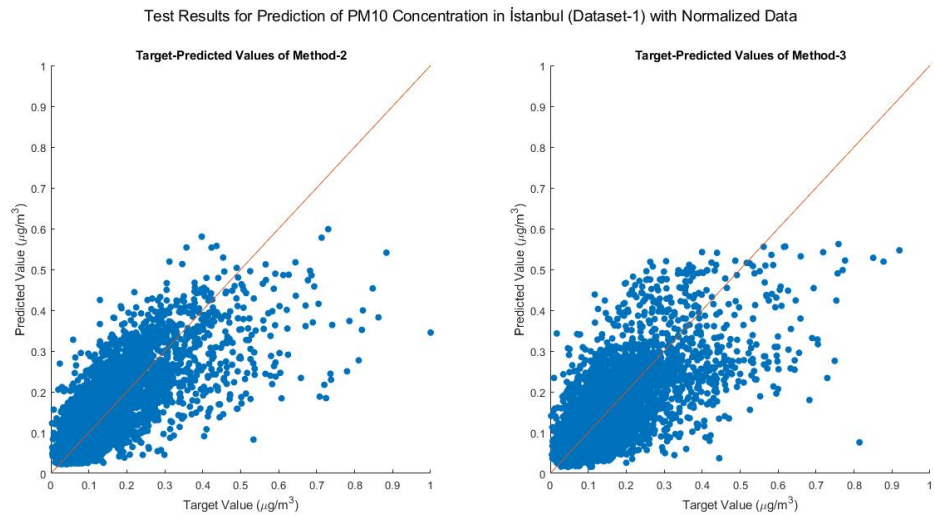


Figure F.3: Scatter plot of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-1) with normalized data

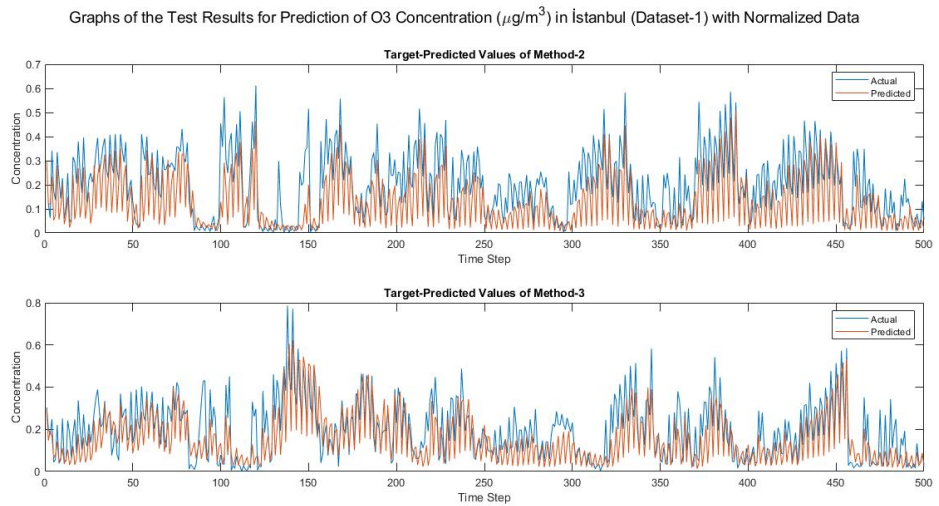


Figure F.4: Graphs of the target-predicted values of Method-2 and Method-3 test results for O_3 in İstanbul (Dataset-1) with normalized data

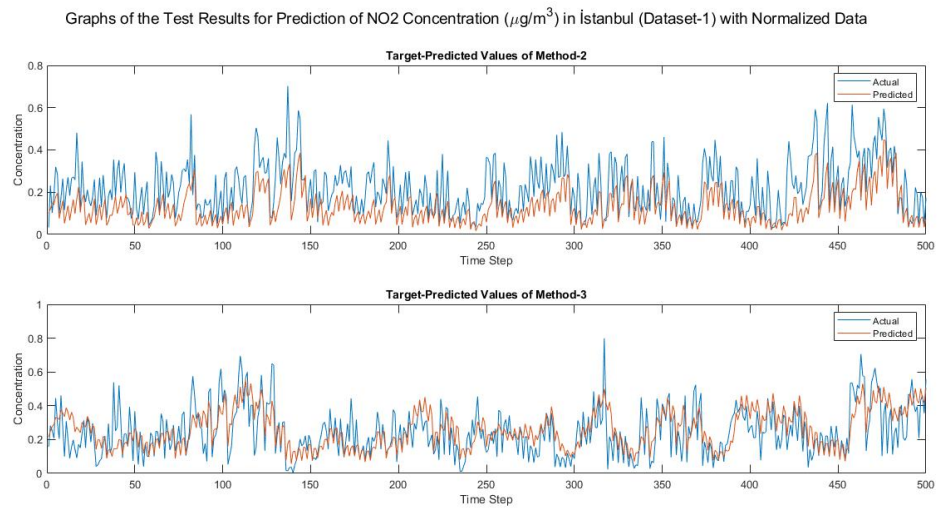


Figure F.5: Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_2 in Istanbul (Dataset-1) with normalized data

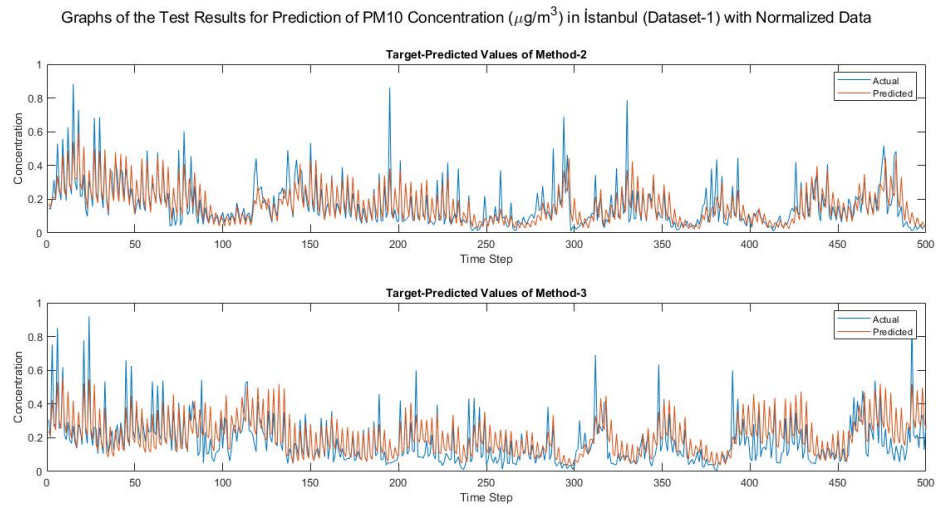


Figure F.6: Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in Istanbul (Dataset-1) with normalized data

APPENDIX G: Results of İstanbul Dataset-2

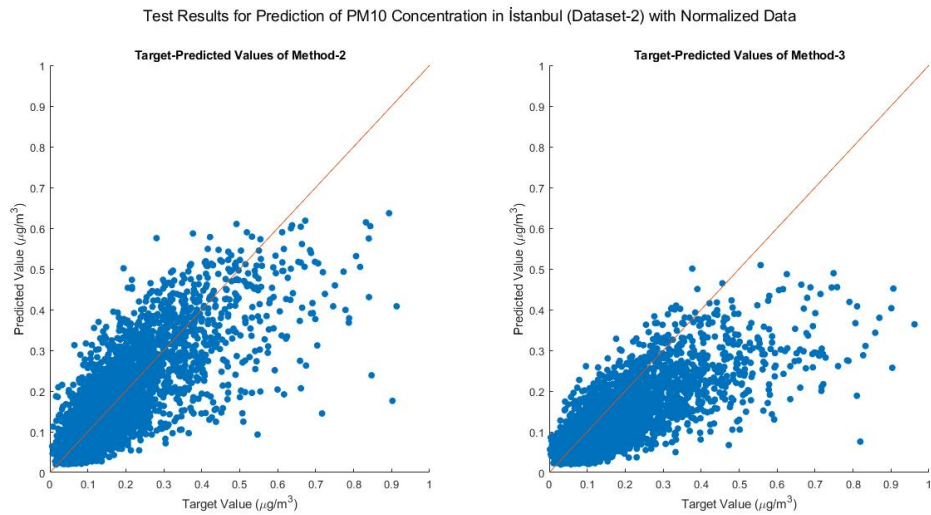


Figure G.1: Scatter plot of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-2) with normalized data

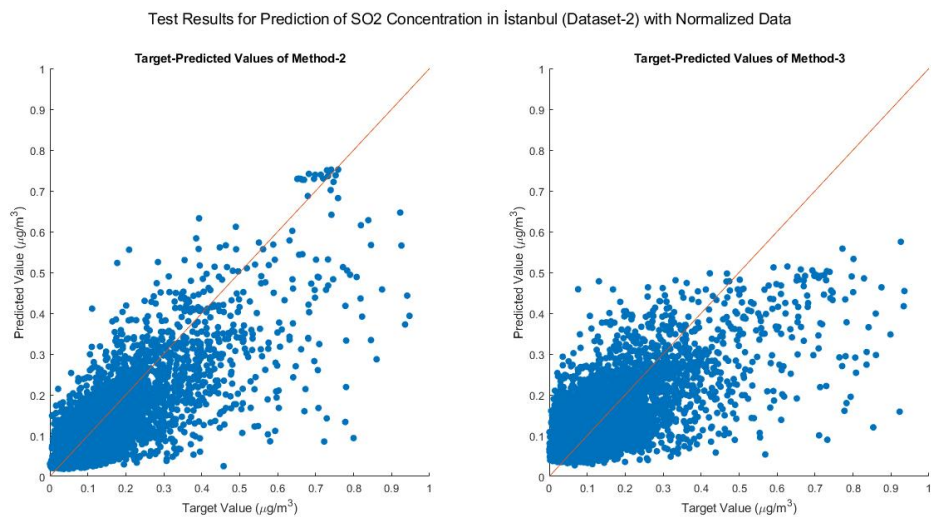


Figure G.2: Scatter plot of Method-2 and Method-3 test results for SO_2 in İstanbul (Dataset-2) with normalized data

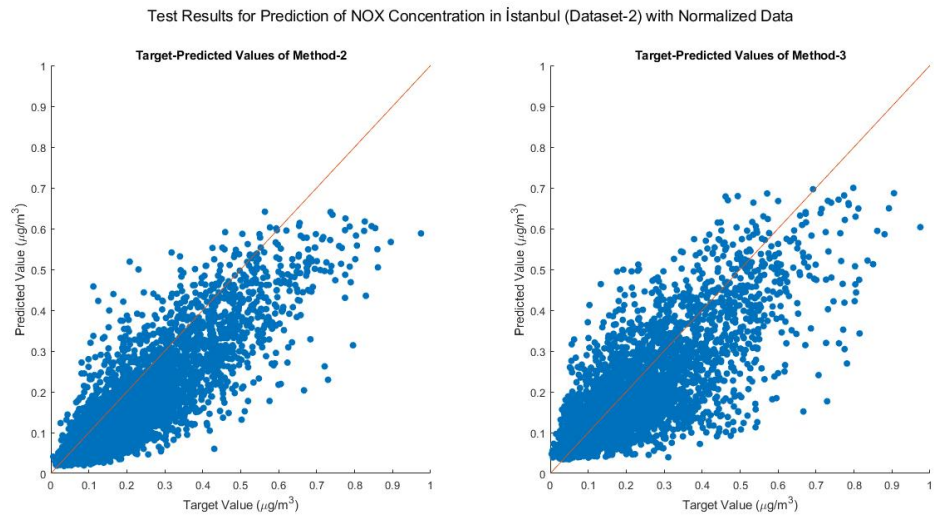


Figure G.3: Scatter plot of Method-2 and Method-3 test results for NO_X in İstanbul (Dataset-2) with normalized data

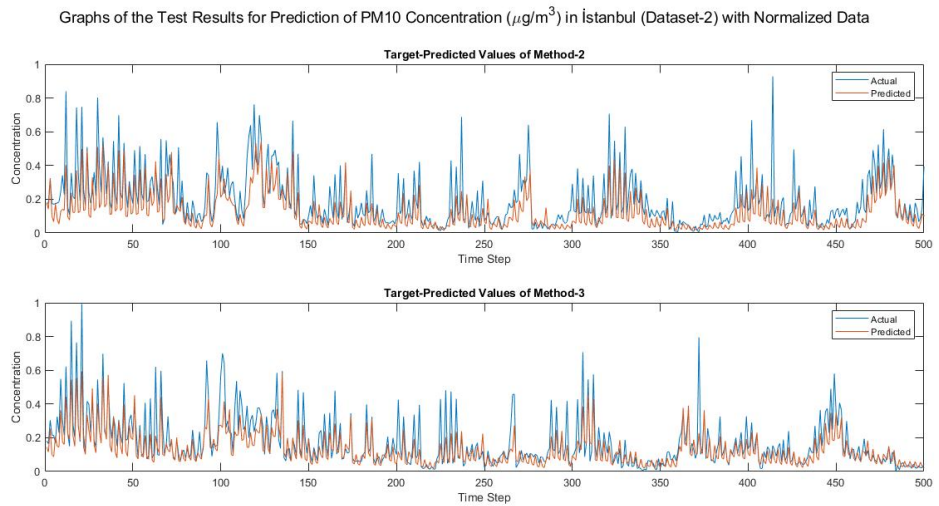


Figure G.4: Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-2) with normalized data

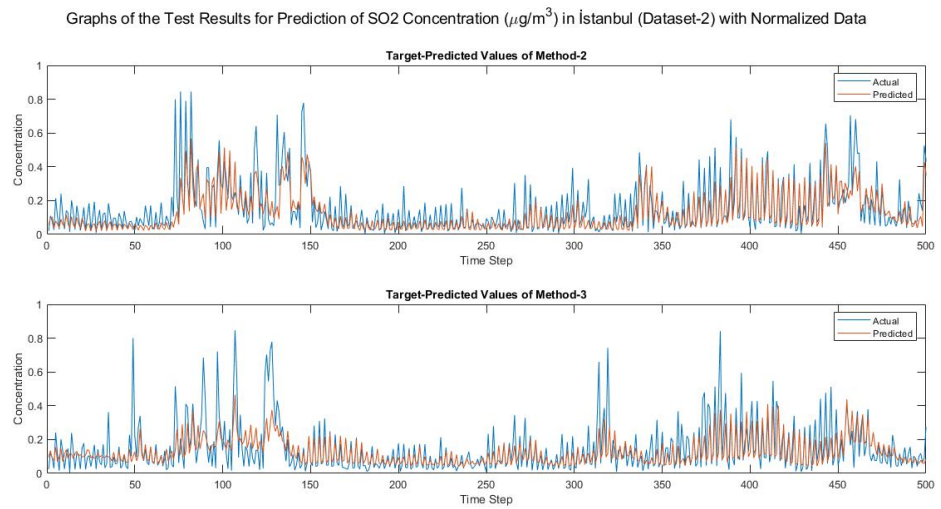


Figure G.5: Graphs of the target-predicted values of Method-2 and Method-3 test results for SO_2 in Istanbul (Dataset-2) with normalized data

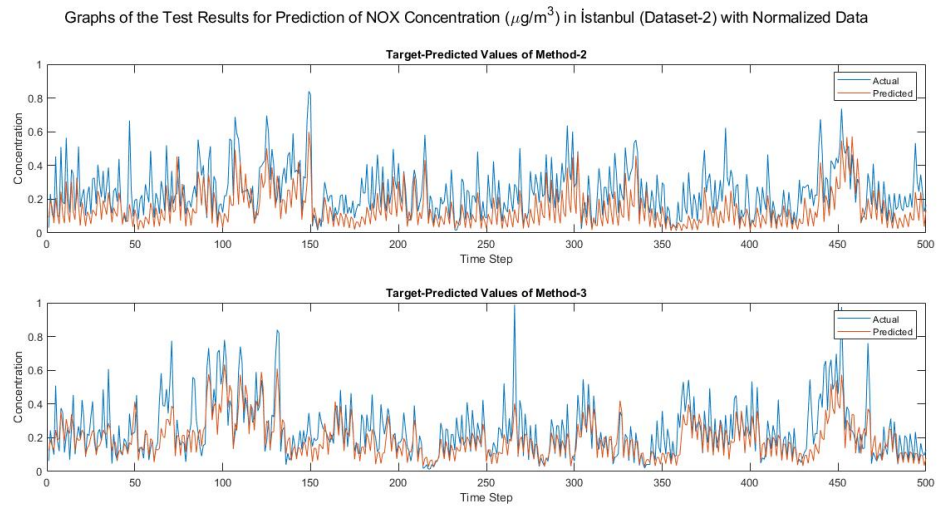


Figure G.6: Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_X in Istanbul (Dataset-2) with normalized data

APPENDIX H: Results with Meteorological Data

H.1 Results for İstanbul Dataset-2

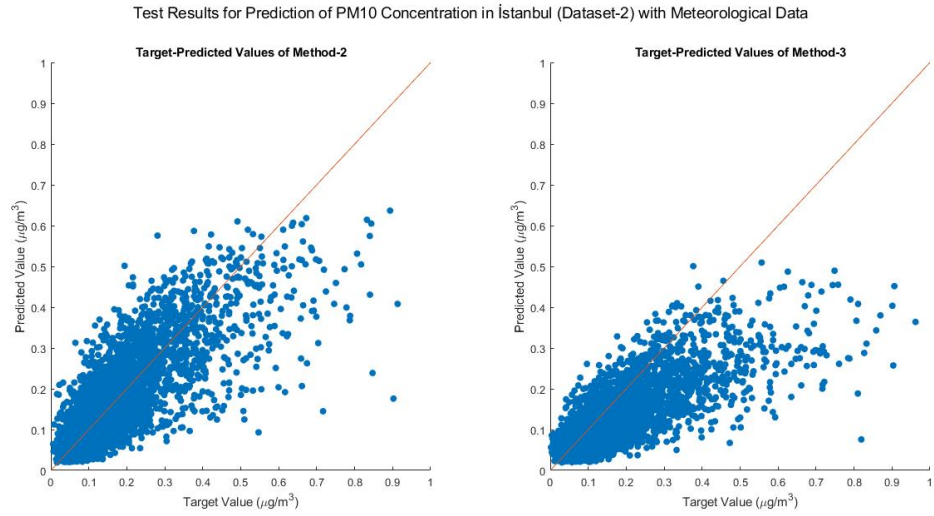


Figure H.1: Scatter plot of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-2) with meteorological data

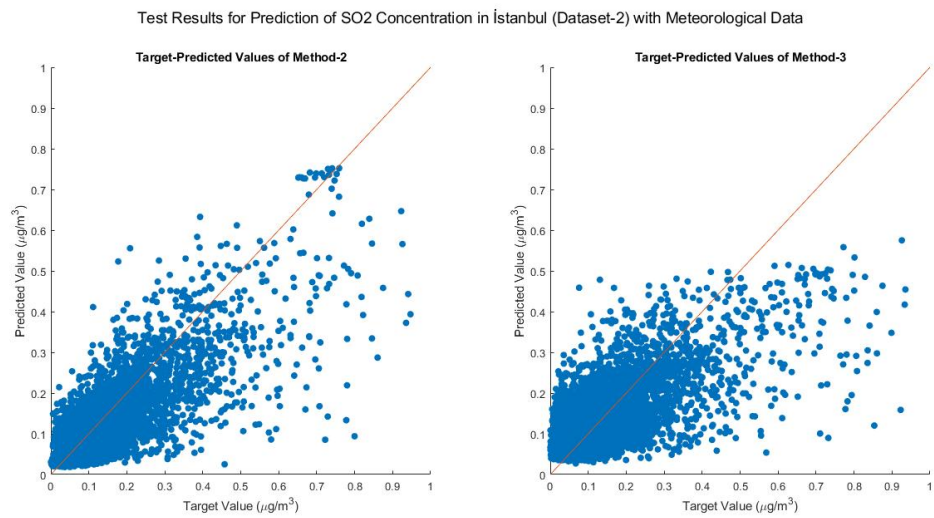


Figure H.2: Scatter plot of Method-2 and Method-3 test results for SO_2 in İstanbul (Dataset-2) with meteorological data

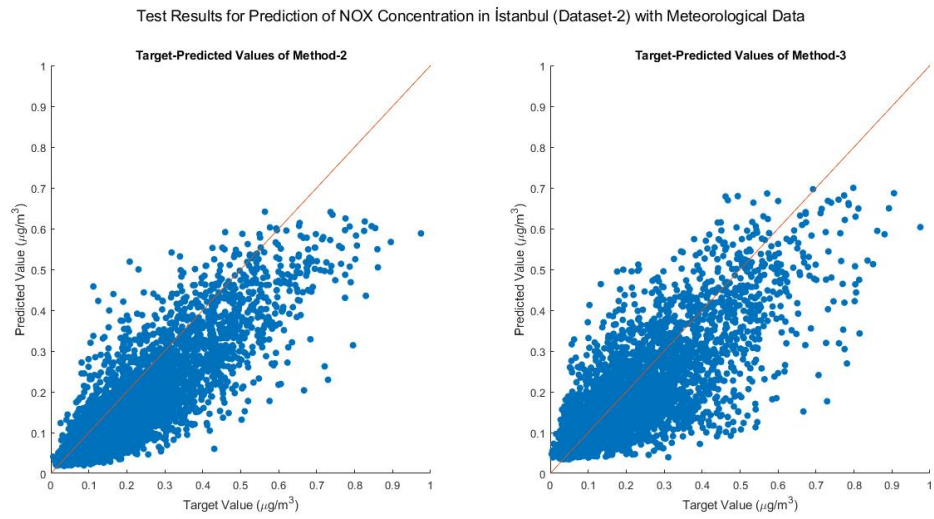


Figure H.3: Scatter plot of Method-2 and Method-3 test results for NO_X in İstanbul (Dataset-2) with meteorological data

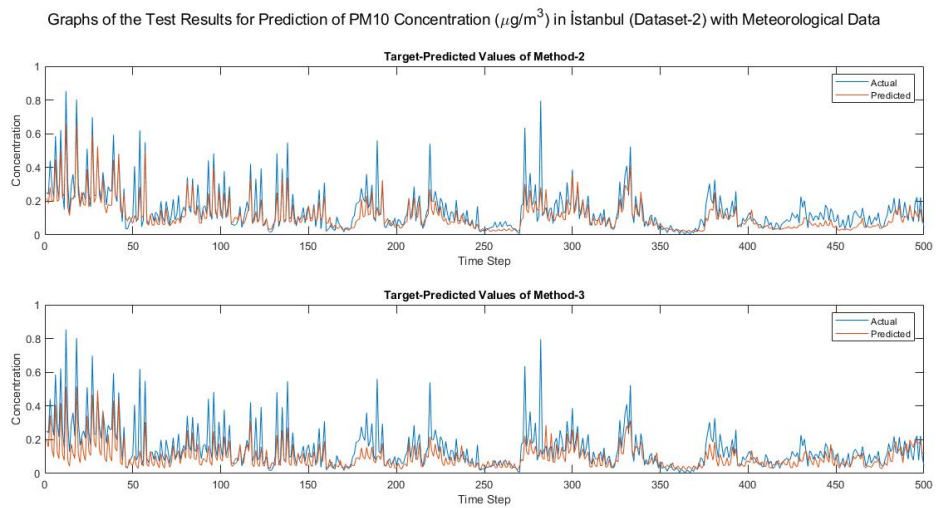


Figure H.4: Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in İstanbul (Dataset-2) with meteorological data

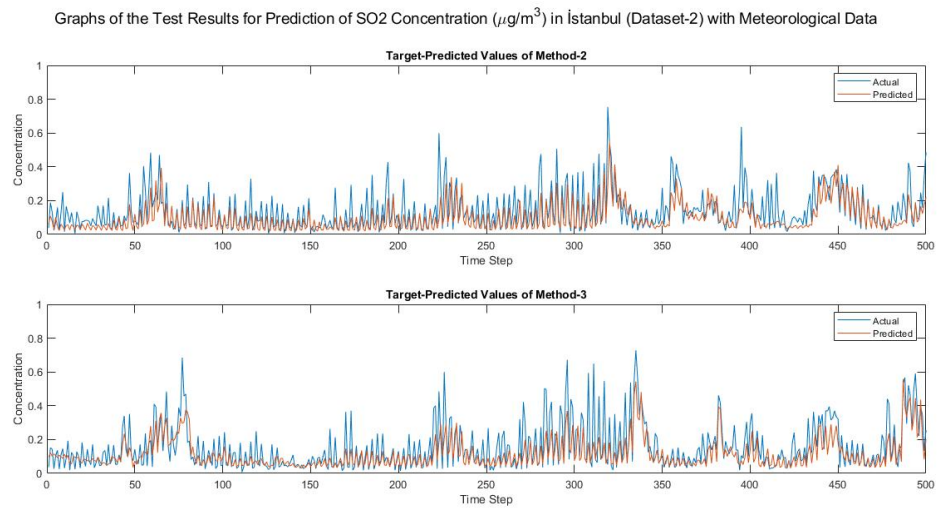


Figure H.5: Graphs of the target-predicted values of Method-2 and Method-3 test results for SO_2 in İstanbul (Dataset-2) with meteorological data

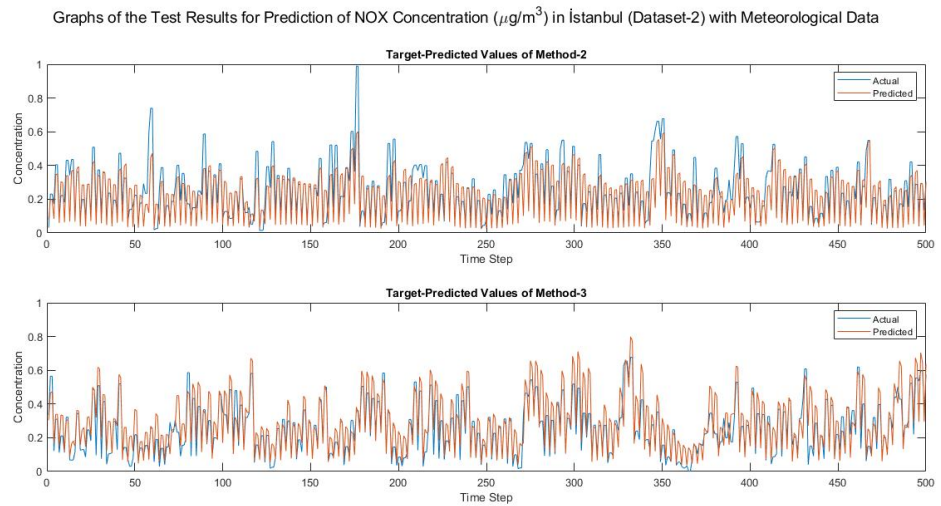


Figure H.6: Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_X in İstanbul (Dataset-2) with meteorological data

H.2 Results for Kocaeli

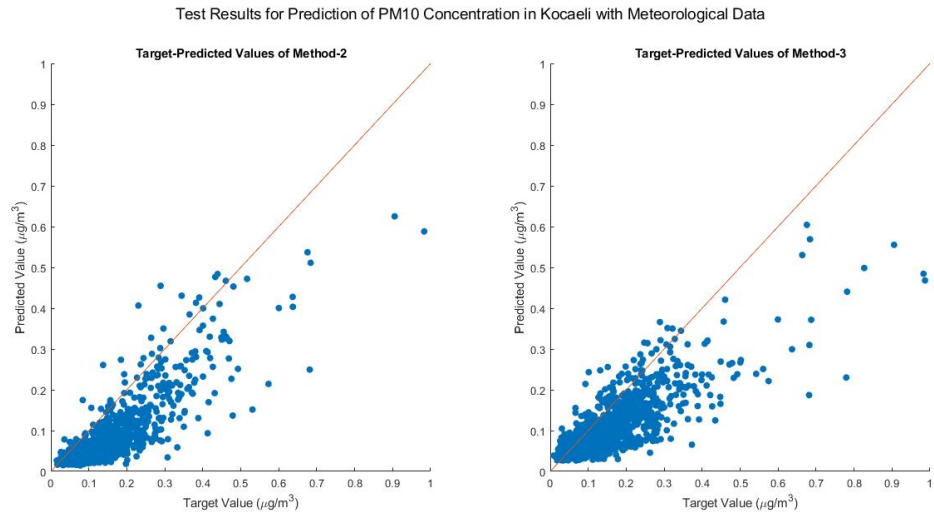


Figure H.7: Scatter plot of Method-2 and Method-3 test results for PM_{10} in Kocaeli with meteorological data

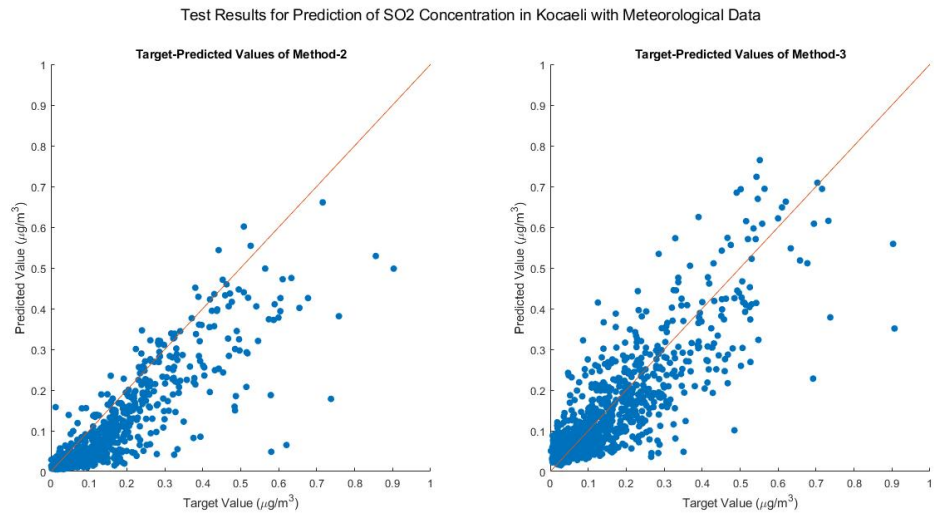


Figure H.8: Scatter plot of Method-2 and Method-3 test results for SO_2 in Kocaeli with meteorological data

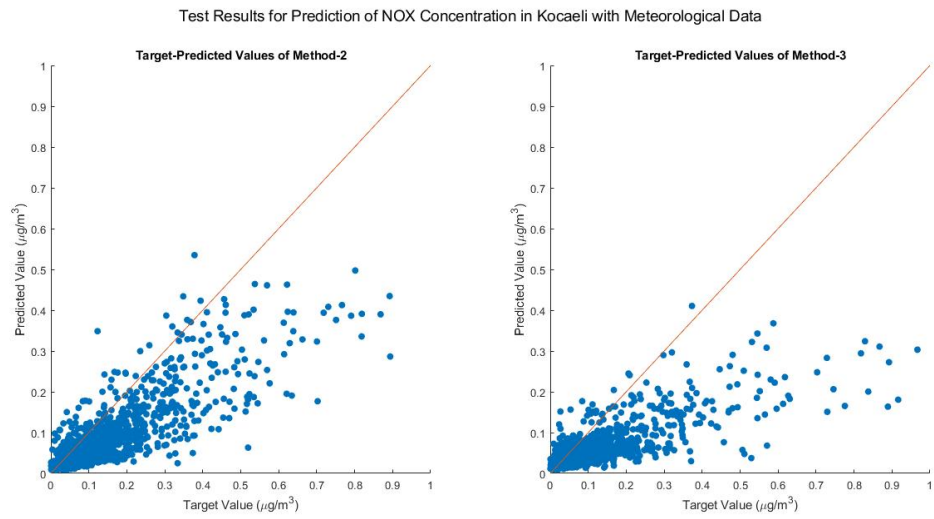


Figure H.9: Scatter plot of Method-2 and Method-3 test results for NO_X in Kocaeli with meteorological data

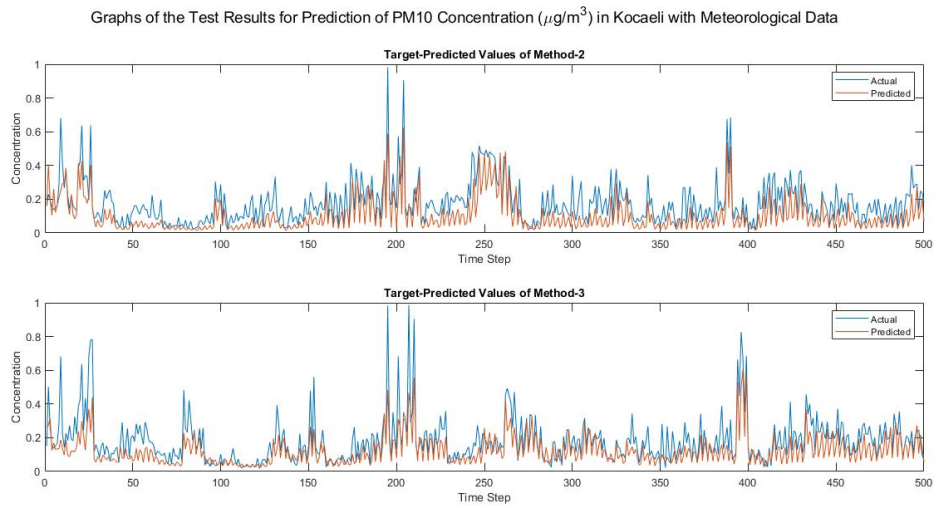


Figure H.10: Graphs of the target-predicted values of Method-2 and Method-3 test results for PM_{10} in Kocaeli with meteorological data

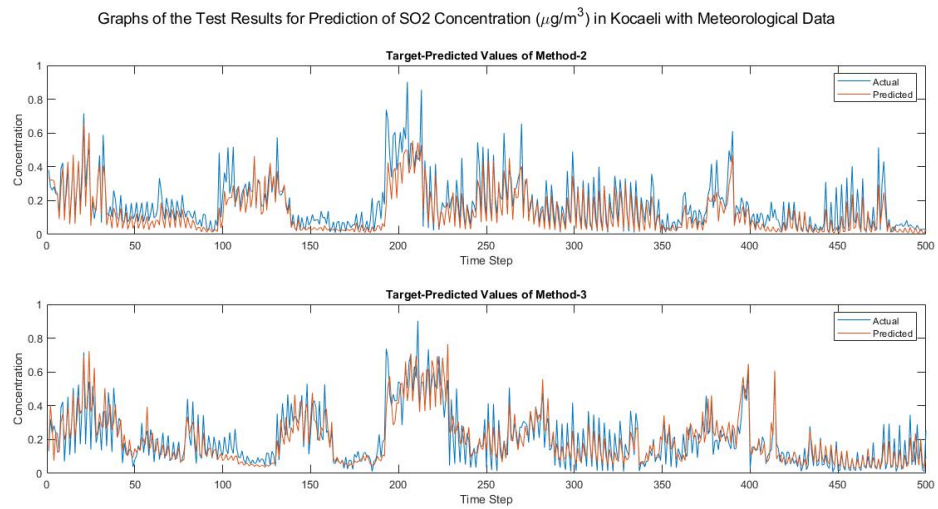


Figure H.11: Graphs of the target-predicted values of Method-2 and Method-3 test results for SO_2 in Kocaeli with meteorological data

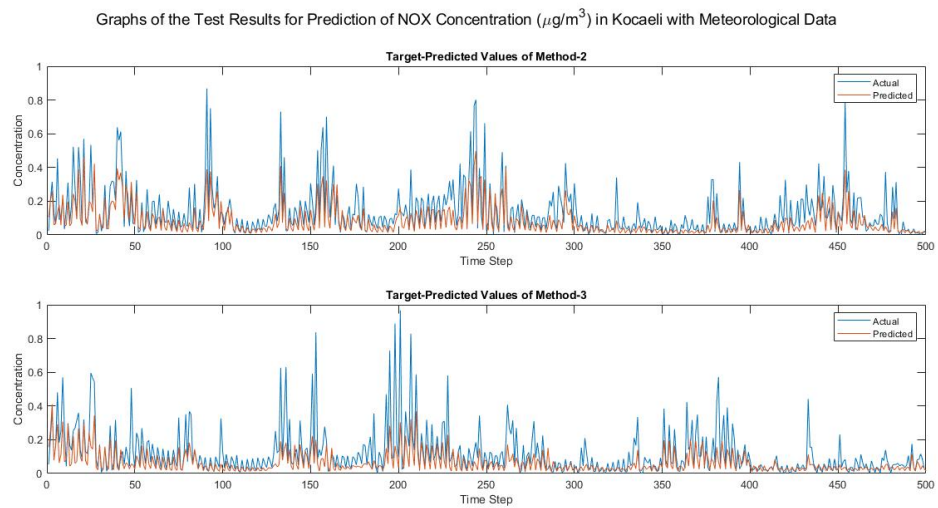


Figure H.12: Graphs of the target-predicted values of Method-2 and Method-3 test results for NO_X in Kocaeli with meteorological data

Graphs of Target-Predicted Values for PM10 in Istanbul (Dataset-2) with Transfer Learning

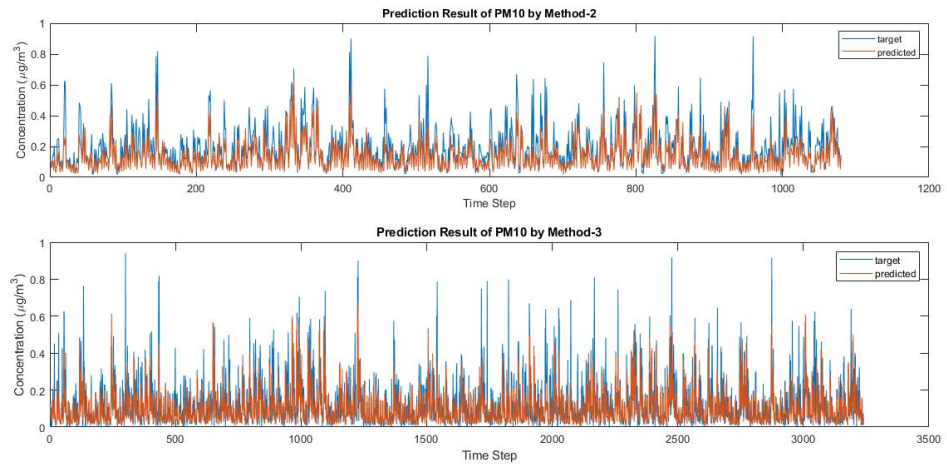


Figure H.13: Graphs of the target-predicted values for PM_{10} in Istanbul (Dataset-2) with transfer learning

REFERENCES

1. N. Scovronick, Reducing global health risks through mitigation of short-lived climate pollutants.
2. E. G. Ortiz, Air quality in europe 2019 report, Tech. rep.
3. 2030 climate & energy framework.
URL https://ec.europa.eu/clima/policies/strategies/2030_en
4. A. G. Ortiz, Air quality in Europe 2019 report, 2019.
5. WHO, Healthy environments for healthier populations: Why do they matter, and what can we do?
6. D. McLaren, J. Agyeman, Sharing Cities: a case for truly smart and sustainable cities, 2015.
7. S. Musa, Smart cities-a road map for development, IEEE Potentials 37 (2) (2018) 19–23.
8. S. Pellicer, G. Santa, A. L. Bleda, R. Maestre, A. J. Jara, A. G. Skarmeta, A global perspective of smart cities: A survey, in: 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2013, pp. 439–444.
9. Smart cities: background paper (2013).
10. The marketplace of the european innovation partnership on smart cities and communities.
URL <https://eu-smartcities.eu>
11. Q. Chen, W. Wang, F. Wu, S. De, R. Wang, B. Zhang, X. Huang, A survey on an emerging area: Deep learning for smart city data, IEEE Transactions on Emerging Topics in Computational Intelligence 3 (5) (2019) 392–410.
12. C. A. Medina, M. R. Prez, L. C. Trujillo, Iot paradigm into the smart city vision: A survey, in: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017, pp. 695–704.
13. D. Evans, How the next evolution of the internet is changing everything, 2011.

14. A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, *IEEE Internet of Things Journal* 1 (1) (2014) 22–32.
15. Y. Lv, Y. Duan, W. Kang, Z. Li, F. Wang, Traffic flow prediction with big data: A deep learning approach, *IEEE Transactions on Intelligent Transportation Systems* 16 (2) (2015) 865–873.
16. G. Pan, G. Qi, W. Zhang, S. Li, Z. Wu, L. T. Yang, Trace analysis and mining for smart cities: issues, methods, and applications, *IEEE Communications Magazine* 51 (6) (2013) 120–126.
17. U. Varshney, Pervasive healthcare and wireless health monitoring, *Mobile Networks and Applications* 12 (2007) 113–127.
18. R. Miotto, L. Li, B. Kidd, Deep patient: An unsupervised representation to predict the future of patients from the electronic health records, *Scientific Reports* 6 (2016) 26094. doi:10.1038/srep26094.
19. B. Ong, K. Sugiura, K. Zettsu, Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting pm2.5, *Neural Computing and Applications* 27. doi:10.1007/s00521-015-1955-3.
20. Z. Qi, T. Wang, G. Song, W. Hu, X. Li, Z. Zhang, Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality, *IEEE Transactions on Knowledge and Data Engineering* 30 (12) (2018) 2285–2297.
21. 2018 revision of world urbanization prospects (2019).
22. L. Snchez, L. Muoz, J. Galache, P. Sotres, J. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, D. Pfisterer, Smartsantander: Iot experimentation over a smart city testbed, *Computer Networks*-doi:10.1016/j.bjp.2013.12.020.
23. K. Kloeckl, O. Senn, C. Ratti, Enabling the real-time city: Live singapore!, *Journal of Urban Technology* 19. doi:10.1080/10630732.2012.698068.
24. T. Bakici, E. Almirall, J. Wareham, A smart city initiative: The case of barcelona, *Journal of the Knowledge Economy* 4. doi:10.1007/s13132-012-0084-9.
25. M. V. Moreno, F. Terroso-Senz, A. Gonzalez-Vidal, M. Valds-Vela, A. F. Skarmeta, M. A. Zamora, V. Chang, Applicability of big data techniques to

- smart cities deployments, *IEEE Transactions on Industrial Informatics* 13 (2) (2017) 800–809.
26. L. Ang, K. P. Seng, A. M. Zungeru, G. K. Ijamaru, Big sensor data systems for smart cities, *IEEE Internet of Things Journal* 4 (5) (2017) 1259–1271.
 27. S. Park, M. Kim, M. Kim, H.-G. Namgung, K.-T. Kim, K. Cho, S.-B. Kwon, Predicting pm 10 concentration in seoul metropolitan subway stations using artificial neural network (ann), *Journal of Hazardous Materials* 341. doi:10.1016/j.jhazmat.2017.07.050.
 28. A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, *IEEE Internet of Things Journal* 1 (1) (2014) 22–32.
 29. P. Soh, J. Chang, J. Huang, Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations, *IEEE Access* 6 (2018) 38186–38199.
 30. G. Zhao, G. Huang, H. He, Q. Wang, Innovative spatial-temporal network modeling and analysis method of air quality, *IEEE Access* 7 (2019) 26241–26254.
 31. I. Djalalova, L. Delle Monache, J. Wilczak, Pm2.5 analog forecast and kalman filter post-processing for the community multiscale air quality (cmaq) model, *Atmospheric Environment* 108. doi:10.1016/j.atmosenv.2015.02.021.
 32. H.-J. Chu, C.-Y. Lin, C.-j. Liao, Y.-M. Kuo, Identifying controlling factors of ground-level ozone levels over southwestern taiwan using a decision tree, *Atmospheric Environment* 60 (2012) 142152. doi:10.1016/j.atmosenv.2012.06.032.
 33. A. Tai, L. Mickley, D. Jacob, Correlations between fine particulate matter (pm2.5) and meteorological variables in the united states: Implications for the sensitivity of pm2.5 to climate change, *Atmospheric Environment* 44 (2010) 3976–3984. doi:10.1016/j.atmosenv.2010.06.060.
 34. K. Bashir Shaban, A. Kadri, E. Rezk, Urban air pollution monitoring system with forecasting models, *IEEE Sensors Journal* 16 (8) (2016) 2598–2606.
 35. Y. Zhang, Y. Wang, M. Gao, Q. Ma, J. Zhao, R. Zhang, Q. Wang, L. Huang, A predictive data feature exploration-based air quality prediction approach, *IEEE Access* 7 (2019) 30732–30743.
 36. S. Khan, D. Paul, P. Momtahan, M. Aloqaily, Artificial intelligence frame-

- work for smart city microgrids: State of the art, challenges, and opportunities, in: 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), 2018, pp. 283–288.
37. D. Qin, J. Yu, G. Zou, R. Yong, Q. Zhao, B. Zhang, A novel combined prediction scheme based on cnn and lstm for urban pm2.5 concentration, *IEEE Access* 7 (2019) 20050–20059.
 38. Q. Tao, F. Liu, Y. Li, D. Sidorov, Air pollution forecasting using a deep learning model based on 1d convnets and bidirectional gru, *IEEE Access* 7 (2019) 76690–76698.
 39. J. Ma, Y. Ding, V. J. L. Gan, C. Lin, Z. Wan, Spatiotemporal prediction of pm2.5 concentrations at different time granularities using idw-blstm, *IEEE Access* 7 (2019) 107897–107907.
 40. Q. Zhang, X. Jiang, D. Tong, S. Davis, H. Zhao, G. Geng, T. Feng, B. Zheng, Z. Lu, D. Streets, R. Ni, M. Brauer, A. Donkelaar, R. Martin, H. Huo, Z. Liu, D. Pan, H. Kan, Y. Yan, D. Guan, Transboundary health impacts of transported global air pollution and international trade, *Nature* 543. doi:10.1038/nature21712.
 41. G. Corani, M. Scanagatta, Air pollution prediction via multi-label classification, *Environmental Modelling and Software* 80 (2016) 259–264. doi:10.1016/j.envsoft.2016.02.030.
 42. H.-L. Yu, W. Chih-Hsin, Retrospective prediction of intraurban spatiotemporal distribution of pm2.5 in taipei, *Atmospheric Environment - ATMOS ENVIRON* 44 (2010) 3053–3065. doi:10.1016/j.atmosenv.2010.04.030.
 43. A. N. Averkin, S. Yarushev, Hybrid approach for time series forecasting based on anfis and fuzzy cognitive maps, in: 2017 XX IEEE International Conference on Soft Computing and Measurements (SCM), 2017, pp. 379–381.
 44. M. Zeinalnezhad, A. Gholamzadeh Chofreh, F. Goni, J. Klemes, Air pollution prediction using semi-experimental regression model and adaptive neuro-fuzzy inference system, *Journal of Cleaner Production* (2020) 121218doi:10.1016/j.jclepro.2020.121218.
 45. D. Schrholtz, S. Kubler, A. Zaslavsky, Artificial intelligence-enabled context-

- aware air quality prediction for smart cities, *Journal of Cleaner Production* (2020) 121941doi:10.1016/j.jclepro.2020.121941.
46. M. Aceves-Fernandez, R. Domnguez-Guevara, J. C. Pedraza Ortega, J. Vargas-Soto, Evaluation of key parameters using deep convolutional neural networks for airborne pollution (pm10) prediction, *Discrete Dynamics in Nature and Society* 2020 (2020) 1–14. doi:10.1155/2020/2792481.
 47. L. Di Antonio, A. Rosato, V. Colaiuda, A. Lombardi, B. Tomassetti, M. Panella, Multivariate prediction of pm 10 concentration by lstm neural networks, 2019, pp. 423–431. doi:10.1109/PIERS-Fall48861.2019.9021929.
 48. M. Lv, Y. Li, L. Chen, T. Chen, Air quality estimation by exploiting terrain features and multi-view transfer semi-supervised regression, *Information Sciences* 483. doi:10.1016/j.ins.2019.01.038.
 49. Y. Wei, Y. Zheng, Q. Yang, Transfer knowledge between cities, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 16*, Association for Computing Machinery, New York, NY, USA, 2016, p. 19051914. doi:10.1145/2939672.2939830.
URL <https://doi.org/10.1145/2939672.2939830>
 50. J. Ma, J. Cheng, C. Lin, Y. Tan, J. Zhang, Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques, *Atmospheric Environment* 214 (2019) 116885. doi:10.1016/j.atmosenv.2019.116885.
 51. E. Eessaar, The database normalization theory and the theory of normalized systems: Finding a common ground, *Baltic Journal of Modern Computing* 4 (2016) 5–33.
 52. Open data bcn, barcelona’s city hall open data service.
 53. Ulusal hava kalitesi izleme ađı (uhkia), s¼rekli izleme merkezi (sim).
URL <https://sim.csb.gov.tr/>
 54. E. Zivot, J. Wang, *Rolling Analysis of Time Series*, 2003, pp. 299–346. doi:10.1007/978-0-387-21763-5_9.
 55. D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Learning Representations by Back-Propagating Errors*, MIT Press, Cambridge, MA, USA, 1988, p. 696699.

56. Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, *IEEE Transactions on Neural Networks* 5 (2) (1994) 157–166.
57. J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence: A survey, *Knowledge-Based Systems* 80. doi:10.1016/j.knosys.2015.01.010.
58. S. J. Pan, Q. Yang, A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering* 22 (10) (2010) 1345–1359.
59. K. Weiss, T. Khoshgoftaar, D. Wang, A survey of transfer learning, *Journal of Big Data* 3. doi:10.1186/s40537-016-0043-6.

