A

Thesis Report

On

Spatiotemporal prediction of dynamic particulate matter using deep learning methods

By

Vineet Gupta

21566020

Under the Guidance of

Prof. Amit Agarwal

Department of Civil Engineering

IIT Roorkee

Mehta Family School of Data Science and Artificial Intelligence

Indian Institute of Technology Roorkee

June 2023

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **"Spatiotemporal prediction of dynamic particulate matter using deep learning methods"** is presented in the partial fulfilment of the requirements for the award of degree of "Master of Technology" with specialization in Data Science, submitted to the Mehta Family School of Data Science and Artificial Intelligence, Indian Institute of Technology Roorkee, under the guidance of **Dr. Amit Agarwal** Department of Civil Engineering and Joint Faculty Mehta Family School of Data science and Artificial Intelligence IIT Roorkee. The matter embodied in this dissertation has not been submitted for the award of any other degree.

Dr. Amit Agarwal
Assistant Professor
Department of Civil Engineering
Joint Faculty Mehta Family School of Data science and Artificial Intelligence
Indian Institute of Technology, Roorkee

Vineet Gupta
21566020

Date: June 02, 2023

Acknowledgement

I would like to express my genuine gratitude and thanks to my respected supervisor **Dr. Amit Agarwal**, Assistant professor, Department of Civil Engineering, Indian Institute of Technology Roorkee, for his valuable guidance and consistent encouragement throughout the work. I am very thankful to him for his kind and useful suggestions and valuable time during the work period. I am also very grateful to **Ms. Rashmi Choudhary**, Research Scholar, Department of Civil Engineering, Indian Institute of Technology Roorkee, for helping with the workflow and mentoring me. She provided valuable help in reviewing the report and presenting various ideas. I would also like to thank **Mr. Vikram Singh**, Research Scholar, Department of Civil Engineering, Indian Institute of Technology Roorkee, for his support. I also extend my heartfelt thanks to my family and friends for their good wishes.

Vineet Gupta
(**21566020**)

Abstract

With the unprecedented increase in the world population, there is an increase in the number of vehicles and industries, which results in a worldwide increase in air pollution. Currently, the world population touched the mark of 8 billion. Vehicular emission is the main reason for the increase in air pollution. Thus, measuring and modeling air pollution and taking preventive actions efficiently becomes extremely important. For modeling, traffic characteristics including traffic volume and density near fixed monitoring sites, play an important role. Other factors such as meteorological data like Relative Humidity (RH), Atmospheric Temperature (AT), Wind Speed (WS), and Barometric Pressure (BP) are also used. The pollution decreases during summers as the temperature increases, wind speed increases in summers, and the humidity is less. Due to all these reasons air pollution decreases as winds blow away the pollution. As the winter approaches at the start of November, the pollutants accumulate in the air due to high humidity and wind speeds. There are only 36 monitoring stations located in Delhi. There is a large need of monitoring stations, but it is not feasible to install new monitoring stations as there is a high cost of setting and maintenance of the static stations. So, there is a need to develop a prediction model for the prediction of the pollutants at the locations that are away from the monitoring stations. The study mainly takes four types of features for model development. These features are: Meteorological features, Traffic flow features, Point of Interest (POI) features, and historical data on pollution gathered from the monitoring stations. The research study aims to develop a model to minimize the error between the actual values and the predicted values.

Keywords: Air Pollution, Spatial Prediction, Temporal Prediction, Monitoring stations

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Air pollution is a global issue that demands attention due to its profound impacts on human health and the environment. It occurs when elements that are hazardous to people and other living things are released into the atmosphere. Pollutants are toxic solids, liquids, or gases that harm our environment and are created in more significant quantities than usual (Manisalidis et al., 2020). It has been known that long-term exposure to toxic air components has many harmful effects on human health. It will also lead to cardiovascular problems and breathing problems. Exposure to polluted air poses risks to human well-being and leads to significant economic burdens in terms of healthcare costs and productivity losses. Moreover, air pollution contributes to climate change by increasing greenhouse gas concentrations, exacerbating the global warming phenomenon and its associated consequences. The main reason for air pollution is burning fossil fuels, industries, increasing traffic, etc. The most common air pollutants are: (a) Solid Particles (Mercury, lead), (b) Particulate matters ($PM_{2.5}$ and PM_{10}), (c) Gaseous Pollutants (NO, CO, SO2)

These could consist of biological or inorganic material. The health of people and other living things may be harmed by these contaminants. According to a recent study from the University of Chicago, air pollution is currently reducing people's life expectancy in Delhi by 10 years. According to the researchers, the Air Quality Life Index demonstrates that particulate matter pollution lowers life expectancy more than communicable diseases. The average Indian will see a five-year reduction in life expectancy as a result of the Indian Ganges plain being the most polluted region in the world and failing to fulfil WHO standards for air quality. High levels of air pollution contribute to diseases including asthma and various pulmonary illnesses. According to a study, chronic exposure to air

pollution appears to cause diabetes (Eze et al., 2014). According to epidemiological and toxicological research, air pollution enters the respiratory tract through the lungs and builds up in lung cells, which negatively impacts heart and lung function and causes other difficulties. Long term air pollution exposure can also lead to change in count of total blood cells (Manisalidis et al., 2020).

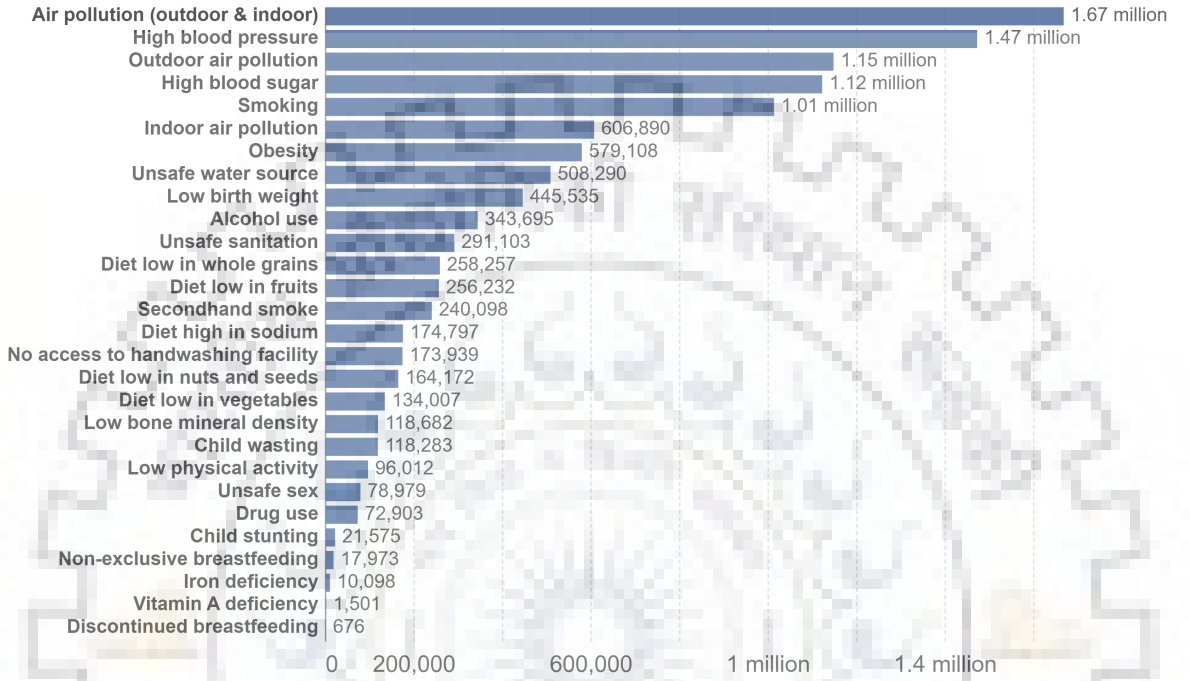

Figure 1.1: Number of deaths by risk factors, India, 2019 Adapted from (Data, 2019)

We can see from the Fig. 1.1 that in India, the highest number of deaths (1.67 Million) in 2019 were due to air pollution (outside and indoor). According to WHO data, nine out of ten people breathe filthy air. The WHO estimates that exposure to particulate matter causes about 7 million deaths annually. $PM_{2.5}$, an extremely fine particulate matter, can enter the lungs deeply and interact with blood to cause diseases like heart attacks, lung cancer, and pulmonary infections like asthma and pneumonia. More individuals are impacted by PM than by any other pollutant. PM's main ingredients are sulfate, nitrates, ammonia, sodium chloride, black carbon, mineral particles, and water. It comprises a complex combination of suspended in the air, solid and liquid particles of organic and inorganic materials. While particles with a diameter of 10 millimeters or less, or PM_{10}, can enter and lodge deep inside the lungs, particles with a diameter of 2.5 microns or less, or $PM_{2.5}$, are much more harmful to human health. The lung barrier can be breached by $PM_{2.5}$, allowing it to enter the bloodstream. Most air pollution-related deaths occur in developing and impoverished nations, primarily in African and Asian nations (WHO,

2008-09). The air pollution in Southeast Asia was five times higher than the WHO standard.

Population movement, which began the urbanization process, triggers economic growth and changes in land usage (Wang et al., 2019). Scholars have also seen a common pattern that the urbanization of people is increasing the concentrations of hazardous air pollution (Larkin et al., 2016). Studies have been conducted to show a link between air pollution and the density of metropolitan populations (Wang et al., 2020). Cars are responsible for around 25% of particle pollution emissions. The amount of air pollution should grow as the number of vehicles on the road increases (Larkin et al., 2016). Currently, air pollution affects the capital city of India and the surrounding areas annually. Predicting air pollution has become an important study area with several key objectives. Accurate prediction models offer researchers a means to comprehend the intricate interactions between different pollutants, meteorological conditions, and geographical factors. By analyzing historical data and utilizing advanced statistical and machine learning techniques, researchers can identify and measure the main contributors to air pollution, such as industry emissions, vehicular traffic, and biomass burning. This knowledge facilitates the development of specific interventions and policies to address pollution sources and minimize exposure risks.

1.2 Need for study

The Asian countries are in the developing stage, particularly India and China. These two countries are the highest populated in the world. Many Asian cities are now among the most polluted in the world because of the rapid industrialization and economic growth that has accompanied them (Chung et al., 2011) . Developed countries such as Europe and America had done their bit of pollution while growing, and now, they are going towards green energy. But on the other side, other developing countries depend highly on coal for their energy supplies. This is the reason for the increasing pollution in the Asian and African regions. The World Health Organization (WHO, 2018) released a report in April 2018 that covered 100 countries over five years from 2011 to 2016 and found that the top 15 most polluted cities by $PM_{2.5}$ concentration were all in Asia, with Delhi at the top of the list among the largest cities in the world (Guttikunda et al., 2019)
.

The measurements are taken by air quality stations that are static. Proper monitoring stations are essential to obtain accurate Air Quality Index(AQI) readings of the

environment. However, it is not practical to build these stations in several locations.

The causes are:

1. It consumes a considerable quantity of space.
2. The cost of establishing a station is around Rs. 15 million, and its annual upkeep is around Rs 2.4 Million per year, which is a considerable cost.

Other solutions, such as crowdsourcing (using mobile phones with sensors), may need to be more reliable Hsieh et al., 2015. Crowdsourcing approaches can only detect limited contaminants, such as carbon dioxide. These technologies cannot effectively quantify significant pollutants such as $PM_{2.5}$ and PM_{10}. Due to the following factors, there is a need for a prediction model that, utilizing monitoring station data, can forecast the concentration of air pollutants at locations without monitoring stations. Researchers worldwide employ various techniques, including interpolation, to estimate the concentration of contaminants. However, the air quality values are not uniform over a region. Several things influence air quality. The three most crucial factors are:

1. Weather circumstances
2. Land usage
3. Traffic

This results in non-smooth numbers, making interpolation procedures challenging to anticipate. Another significant issue is the need for more data. Due to the small number of sites, the spatial data is sparse. We must use temporal data (traffic and weather) and additional parameters such as Point of Interest (POI) data to reduce the error in our prediction.

1.3 Objectives of the study

The aim of this study is to develop a prediction model that can predict the parameter count of the air pollution causing elements using the spatial and temporal features such as meteorological factors, traffic density, POI, and pollutants concentration of the locations with monitors. To carry out this study, report is divided into following parts:

1. To understand various techniques used by researchers to predict the pollution concentration.
2. To develop a model that can predict the pollutants concentration as close as possible to the original data.

1.4 Organization of report

This seminar report includes three chapters and is organised as following:

1. Chapter 1 initiates with introduction of air pollution and the factors responsible for air pollution. Further the objective of the study and need for doing this study is elaborated.
2. Chapter 2 describes the key findings from various literature studies and the factors that are directly or indirectly affecting the modelling function are mentioned.
3. Chapter 3 discusses about what data is needed for modelling and how the data is extracted for the use in modelling. It also talks about data preprocessing and different models used for the predictions.
4. Chapter 4 discusses about the results achieved using different models. It also describe the data analysis part for the processed data.
5. Chapter 5 discusses the conclusion and future work that can be done for further improvements in the results.

Chapter 2

Literature Review

Several predictive modeling research had been done in the past. The main factors that can be used for a good predictive model are historical data on pollutants of the monitoring stations, meteorological data of the locations where the stations are present, and locations where the pollutants need to be predicted. The researchers used basic to all complex models for the prediction and reduced the error in the prediction. The literature review is done to understand all the models and the approaches that had been used by the researchers in the past. To identify articles related to the study, keywords such as 'Air Pollution', 'Spatial Prediction', and 'Temporal Prediction' are used on Google Scholar. The references from the studies are also used for understanding the approach. Only articles from 2001 to 2022 in the English language are used for the study.

Most of the carbon, ion, and chemical components found in urban $PM_{2.5}$ are known to originate from sources connected to traffic, industrial emissions, biomass combustion, and salt combustion (Manojkumar et al., 2021). $PM_{2.5}$ has a higher impact on people's lives since it travels a greater distance and is suspended in the atmosphere for a longer period. The main challenge facing emerging nations like India is $PM_{2.5}$, which contains a variety of harmful components. Various models such as Proximity based assessments, interpolation methods, predictive mapping and spatial temporal DNN are used in the past.

2.1 Models used in the past

This section deals with several other methods that can be used for pollutants prediction for the locations with no monitoring stations. Some of the methods used by researchers are: Proximity based assessments, Interpolation Methods, Predictive Mapping and Spatial temporal DNN. The most important models in the context of the study are described

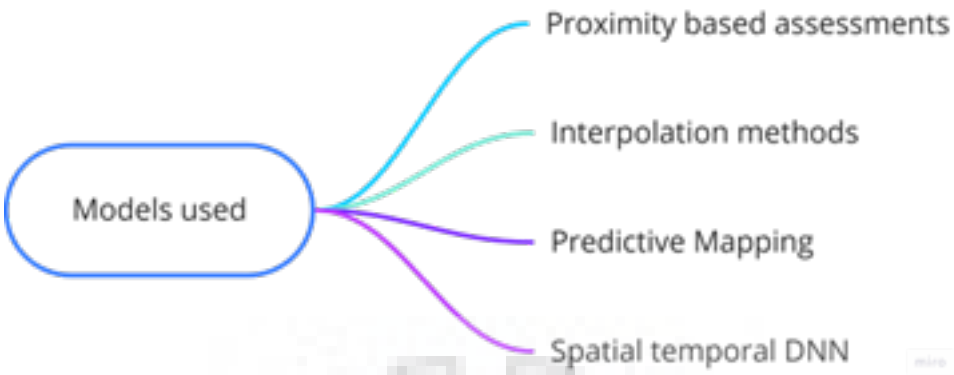

Figure 2.1: Different models used for air pollution prediction

below to determine their characteristics and strong and weak points. Fig. 2.1 shows some of the models that were used in the past.

2.1.1 Proximity based assessments

The most fundamental method for getting the air pollution exposures at any particular location is dependent on subject's proximity to a pollution source. The base of this model is that the closer is the pollution producing sources the more is the pollution at a particular location. A traffic index is used to check the credibility of the model (Venn et al., 2000). Traffic flow near schools were accessed as a continuous measure of traffic density for one kilometre square grid cells. The study found out that the air pollution increases with the increase in the traffic flow.

2.1.2 Interpolation methods

Several interpolation methods such as IDW (Inverse distance Weighted) interpolation and kriging interpolation can be used for the prediction. A network of monitoring stations spread out across the study area is used to measure the target pollutant. The objective is to interpolate results to generate estimates of pollutant concentrations at sites other than monitoring station locations. In IDW interpolation, the inverse distance factor is calculated for each location because as the distance increases, the correlation between the pollutants decreases. Kriging is the most used algorithm for interpolation. By utilising weights that indicate the correlation between the data at two sample sites or between a sample location and the location to be estimated, the regression-based technique known as kriging estimates values at unsampled locations (Diem and Comrie, 2002). It offers fair estimates of values at unsampled places with the least amount of estimated variance.

2.1.3 Predictive mapping

For mapping air pollution in most urban areas, distance-weighting and kriging are often not appropriate methods. The relative scarcity of air quality sensors (small sample size) and their presumably uneven geographical distribution provide the biggest challenges to applying these interpolation algorithms (i.e. inappropriate sampling scheme). A substantial collection of evenly spaced, spatially autocorrelated data is needed for distance-weighting. Contrary to distance-weighting and kriging, linear regression models may generate an accurate surface without the need for spatially autocorrelated observations. Measured observations behave as partially repeated measurements of a single observation rather than as single observations for spatially autocorrelated data. A dependent variable and one or more independent variables are statistically related in a linear regression analysis, which results in an equation. The dependent variable's anticipated values are produced using the equation.

2.1.4 Spatial temporal DNN

This is a modeling technique used to predict the pollutants. In this model both spatial and temporal methods are taken into consideration. Spatial factors are POI and landuse. Temporal factors used are meteorological data such as temperature, RH, wind speed etc. and other factors such as $PM_{2.5}$ and PM 10. The model was trained using historical data per hour for one year (Soh et al., 2018). A technique for integrating relevant data that relies on temporal and spatial correlations between monitoring sites. A convolutional neural network (CNN) and long short-term memory were used to merge many neural network topologies after first identifying the most pertinent spatial-temporal relationships between places (LSTM).

2.2 RAQ : Random forest

In their study, (Yu et al., 2016) proposed a random forest-based approach, called RAQ, for predicting air quality in urban sensing systems. The authors recognized the importance of leveraging data from urban sensing systems to improve air quality predictions. The random forest algorithm was chosen due to its ability to handle complex datasets, capture non-linear relationships, and provide robust predictions. The RAQ approach presented by (Yu et al., 2016) involves collecting air quality data from urban sensing systems, such as sensor networks deployed throughout the city. The collected data, including pollutant concentrations and meteorological variables, are input features for the random

forest model. The authors aimed to develop a predictive model capable of forecasting air quality levels in real time by training the model with historical data. The advantages of the RAQ approach are evident in its ability to handle large and heterogeneous datasets. Random forests can effectively handle missing data and outliers, leading to more reliable predictions. The ensemble nature of the random forest algorithm allows for capturing the interactions and nonlinearities between air quality parameters and meteorological variables, which are essential for accurate predictions. The research by (Yu et al., 2016) contributes to the field of air quality prediction by introducing the RAQ approach, which leverages urban sensing systems and the random forest algorithm. This innovative approach could enhance urban air quality monitoring and management. The RAQ approach can assist policymakers, environmental agencies, and urban planners in making informed decisions to mitigate air pollution and protect public health by providing real-time predictions. Further research could expand the RAQ approach to incorporate additional input features like traffic patterns, landuse data, or emission inventories. Additionally, the performance of the RAQ model can be evaluated and compared with other prediction models to assess its accuracy and effectiveness in different urban environments. In conclusion, the study presents the RAQ approach as a promising method for predicting air quality in urban sensing systems. Random forest modeling allows for robust predictions, capturing complex relationships between air quality parameters and meteorological variables. Applying the RAQ approach can significantly contribute to air pollution management and the development of sustainable urban environments.

2.3 landuse regression

In their comprehensive review, (Hoek et al., 2008) focus on land-use regression (LUR) models as a valuable tool for assessing the spatial variation of outdoor air pollution. LUR models quantify the relationship between air pollution measurements at monitoring sites and land-use characteristics, traffic, and other relevant variables. The authors highlight the importance of LUR models in providing high-resolution spatial air pollution maps, essential for understanding exposure patterns and identifying pollution hotspots. They discuss the critical components of LUR models, including the selection of predictor variables, the modeling approach, and the validation techniques used to assess model performance. (Hoek et al., 2008) provide a detailed overview of the land-use characteristics commonly included in LUR models, such as traffic intensity, industrial emissions, population density, and green space. They also discuss the challenges of selecting ap-

propriate predictor variables and considering temporal variations in air pollution levels. The review highlights the advantages of LUR models, such as their ability to capture small-scale spatial variation and provide insight into the impact of specific land-use factors on air pollution levels. The authors also discuss the limitations of LUR models, including the need for extensive monitoring data, potential biases, and the challenges of extrapolating results to areas without monitoring stations. The research contributes significantly to the field by synthesizing existing knowledge on LUR models for assessing spatial variation in outdoor air pollution. Further research in this area could address the limitations of LUR models, such as improving the spatial representativeness of monitoring sites, considering the influence of meteorological factors, and incorporating new data sources, such as remote sensing data and mobile monitoring technologies. Additionally, exploring the application of LUR models in different geographical regions and investigating their utility for other air pollutants would further enhance the understanding and applicability of these models. (Larkin et al., 2017) address the need for a global-scale landuse regression (LUR) model to estimate NO2 concentrations, considering the importance of understanding the spatial patterns and variability of this pollutant across diverse regions. The authors recognize that LUR models provide a robust approach to predict air pollution levels by incorporating landuse and other relevant spatial predictors. The results of the study indicate that the global LUR model accurately estimates NO2 concentrations across different geographical regions. The model incorporates several important predictors, including landuse, population density, road networks, and satellite-derived data, to account for the major sources and spatial patterns of NO2 pollution. The results of the study indicate that the global LUR model accurately estimates NO2 concentrations across different geographical regions. The model incorporates several important predictors, including landuse, population density, road networks, and satellite-derived data, to account for the major sources and spatial patterns of NO2 pollution. The study presents a comprehensive analysis of NO2 concentrations using a large dataset consisting of satellite-derived NO2 observations, ground monitoring data, and various geospatial predictors. The authors apply a hierarchical modeling framework to develop a global LUR model that captures the spatial variations in NO2 concentrations, accounting for regional differences and the influence of local landuse characteristics. In conclusion, the papers provide a brief review of LUR models for assessing the spatial variation of outdoor air pollution. The authors emphasize the importance of these models in generating high-resolution pollution maps and understanding the role of land-use factors in shaping air quality patterns. The review contributes to advancing the knowledge and application of

LUR models, ultimately aiding in developing targeted air pollution mitigation strategies.

2.4 Spatio temporal prediction

(Wen et al., 2019) address the complex challenge of air pollution prediction by proposing a novel spatiotemporal convolutional long short-term neural network (ST-CLSTM). The authors recognize the need to capture the spatiotemporal relationships between air pollution and meteorological variables, which play an important role in influencing air quality. The study highlights the limitations of traditional modeling approaches in properly capturing the complex patterns and non-linear dynamics of air pollution. To overcome these limitations, the authors introduce the ST-CLSTM model, which combines the strengths of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. This enables the model to learn the spatial and temporal dependencies inherent in air pollution data.

The proposed ST-CLSTM model achieves accurate and reliable air pollution predictions by leveraging the spatiotemporal information in multi-dimensional data. The study demonstrates the superiority of the ST-CLSTM model over conventional models through extensive experiments and comparisons. The results indicate that the ST-CLSTM model outperforms other models in terms of prediction accuracy and generalization capability. The study also highlights the significance of using meteorological variables into air pollution prediction models. Meteorological factors, such as temperature, humidity, wind speed, and atmospheric pressure, have a good influence on air pollution levels. By integrating meteorological data into the ST-CLSTM model, the authors effectively capture the relationship between meteorological conditions and air pollution dynamics.

The application of the ST-CLSTM model holds considerable promise for various practical scenarios. The model's high-resolution predictions offer valuable insights into the spatial and temporal distribution of air pollution, aiding in the identification of pollution hotspots and the formulation of targeted mitigation strategies.

In conclusion, the study conducted by (Wen et al., 2019) introduces a novel spatiotemporal convolutional long short-term neural network (ST-CLSTM) for air pollution prediction. The proposed model successfully addresses the challenges associated with capturing complex spatiotemporal relationships and non-linear dynamics in air pollution data. By integrating meteorological variables into the model, the authors provide a comprehensive understanding of the interdependencies between meteorology and air quality. The ST-CLSTM model demonstrates good performance and offers valuable insights for

Distance	R^2	R^2	Correlation
D3	0.868	C3	0.912
D6	0.898	C6	0.913
D9	0..913	C9	0.919

Table 2.1: R^2 value (Gryech et al., 2021)

air pollution management and decision-making processes.

(Gryech et al., 2021) proposes a novel method for spatial prediction of urban air pollution. The method uses a combination of machine learning and spatial interpolation techniques. The machine learning model is used to learn the relationship between air pollution levels and a set of environmental and meteorological features. The linear interpolation techniques is used to fill in the missing values in the air pollution data. Morocco. The results showed that the method was able to improve the accuracy of spatial prediction of air pollution.

A linear model is used for prediction. In a linear model, Multiple Linear Regression is used. It is used to predict the measurements of the station that is not working (Gryech et al., 2021). The research paper focuses on the prediction of NO2 as this is the only pollutant that's data is available for all the 20 stations. R^2 (R-squared) value is used as a performance metric. After training the model, the R^2 value comes out to be 0.93. We can see that the performance is good, but there is a disadvantage in that they are taking data from all the stations. This will lead to an increase in the model complexity. This model is not robust as it won't work when more than one station breaks down. So, to reduce complexity, they had taken the top 3 or top 6 correlated stations from the correlation matrix and the top 3 or top 6 closest stations from the distance matrix. The author has tried to predict the pollutants of the PA 18 station.

(Gryech et al., 2021) used only the correlation between different stations and the distance between them is taken as the factor to predict the pollutants. Still, there are more such factors that are influencing the concentration of the contaminant. Two of the elements are meteorological and traffic-related factors. Meteorological factors include temperature, wind speed, wind direction, humidity, and pressure in the environment. So, now they have used meteorological and traffic features for air quality prediction. (Gryech et al., 2020) focuses on using meteorological and traffic related features to predict air quality levels. The paper proposes a novel machine learning model that combines a support vector machine (SVM) with a random forest (RF) model. The model was trained and tested on a dataset of air quality data from the city of Casablanca, Morocco. The results showed that the model was able to achieve a high accuracy in predicting air quality

levels. The study by (Samal et al., 2021) addresses the need for accurate spatiotemporal prediction models for air quality, considering the significant impact of air pollution on public health and urban planning. The authors recognize the inherent challenges in capturing the complex spatial and temporal variations of air pollutants, which are influenced by various factors such as emissions, meteorology, and landuse patterns.

The authors suggested a novel strategy that blends deep learning methods and distance-based interpolation approaches to address these issues. By using the recorded values from close-by monitoring stations, distance-based interpolation techniques offer a way to calculate the levels of air pollution at unmonitored locations. Using this interpolation technique, a thorough spatiotemporal air quality dataset may be produced. To further capture the complex correlations and patterns found in the spatiotemporal air quality data, the author used deep learning techniques. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are two deep learning models that are used to extract significant features and understand the complicated temporal correlations contained in the data. The suggested model's accuracy and prediction power are improved by combining deep learning and distance-based interpolation.

The effectiveness of the approach is demonstrated through extensive experiments and evaluations conducted on real-world air quality datasets. The results indicate that the combined distance-based interpolation and deep learning model outperforms traditional interpolation techniques and standalone deep learning models. The proposed methodology achieves high prediction accuracy for both spatial and temporal dimensions, enabling reliable air quality estimates at unobserved locations and future time points.

In conclusion, (Samal et al., 2021) presents a comprehensive study on spatiotemporal prediction of air quality using distance-based interpolation and deep learning techniques. The integration of these approaches addresses the challenges associated with capturing the complex spatial and temporal variations of air pollution. The proposed model offers a valuable tool for accurately predicting air quality at unmonitored locations and future time points, supporting decision-making processes for urban planning and pollution management.

Chapter 3

Methodology

3.1 Overview

The main idea behind this study is to develop a spatio-temporal model for development of a predictive technique for concentration of various pollutants. As discussed in the literature review the air pollution for the locations with no monitors can't be directly predicted using only the historical data of the monitoring stations. Other factors affecting the pollutants at any location are meteorological factors such as wind speed, RH (Relative Humidity), Temperature and pressure. Traffic and road related data is also used as it is observed from the literature review that the air pollution is directly proportional to the traffic flow at any particular area. Other factors such as POI (point of interest) is also an important feature for decreasing the error in our prediction model. A point of interest (POI) is a specific point location that someone may find useful or interesting. It includes the locations such as hotels, schools, colleges, factories, water bodies, forest, parks etc. The area of study is Delhi, India.

3.2 Dataset

Data is the most important aspect for the modelling of the prediction model. The data is scraped using various API's (Application Programmable Interface) available on the internet. Different API's such as HERE maps, Open Street Maps (OSM) etc. are used for the data extraction. The historical data of the pollutants at the locations of monitoring stations and the meteorological data is present in downloadable form at the CPCB (Central Pollution Control Board) and DPCC (Delhi Pollution Control Committee) website. The POI (Point of Interest) data is taken from the OSM (Open Street Maps). QGIS (Quantum Geographic Information System) is used do all the spatial analysis.

3.2.1 Pollution data

The pollution data is the independent variable that is used as a label to predict the pollutants for the study region. It is obtained from the fixed monitoring stations located at various locations in the study region. There are total of 40 monitoring stations located in Delhi region. Out of these there are 24 sites operated by Delhi Pollution Control Committee (DPCC) while 6 sites are managed by CPCB, and rest 6 sites are maintained by India Meteorological Department (IMD) and 2 sites by Indian Institute of Tropical Meteorology (IITM). The real time data is available at Central Control Room for Air Quality Management. The pollutant taken into consideration is $PM_{2.5}$ .The data is taken for six months that is 1 November 2022 to 30 April 2023. The data is taken in an interval of 15 minutes.

3.2.2 Meteorological data

The meteorological data is an important aspect for the modelling of the prediction model. The meteorological data includes Atmospheric Temperature (AT), Wind Speed (WS), Relative Humidity (RH) and Barometric Pressure (BP). All these parameters were obtained from the Central Pollution Control Board's live monitoring stations at 15 minutes interval. The data was obtained as excel file. The following data was recorded:

1. Atmospheric Temperature (AT): The temperature of the Earth's atmosphere at various altitudes is referred to as the "atmospheric temperature." It is influenced by a variety of factors, such as altitude, humidity, and solar energy. The four layers of the atmosphere can be distinguished by the temperature variations that occur at varying heights in relation to the Earth's surface.
2. Wind Speed (WS): Air flows from high pressure to low pressure, primarily as a result of temperature differences, and this process produces wind speed, a fundamental atmospheric quantity. The relationship between wind speed and particle pollution is complex.
3. Relative Humidity (RH): The ratio of the amount of atmospheric moisture that is actually present to the amount that would be contained if the air were saturated is known as relative humidity. Relative humidity depends on both moisture content and temperature because the latter is temperature dependent. The relative humidity is determined using the relevant temperature and dew point for the specified hour.

4. Barometric Pressure (BP): The pressure that results from the weight of the air above us is known as barometric pressure. Despite being relatively light, the air in the atmosphere above us starts to take on some weight as gravity pulls the air molecules together.

3.2.3 Traffic data

The traffic data required for the study is obtained from HERE Maps API. The data was obtained in Extensible Markup language (XML) file format at every 15 minutes interval for six months. The data obtained include speed and congestion for the area of interest. The area of interest can be defined by providing the bounding box. To obtain the data a request is generated and for each request contains information on what data is required. For this study the following request was generated:

`https://traffic.ls.hereapi.com/traffic/6.2/flow.xml?apiKey='+HEREAPIKEY+'&bbox=28.086520,76.730347;28.92163,77.631226&responseattributes=sh,fc`

This request can be divided into subparts as shown in the Fig. 3.1. The response from the request is obtained in xml format which was converted to ordered dictionary in python and the objects from the results are extracted.

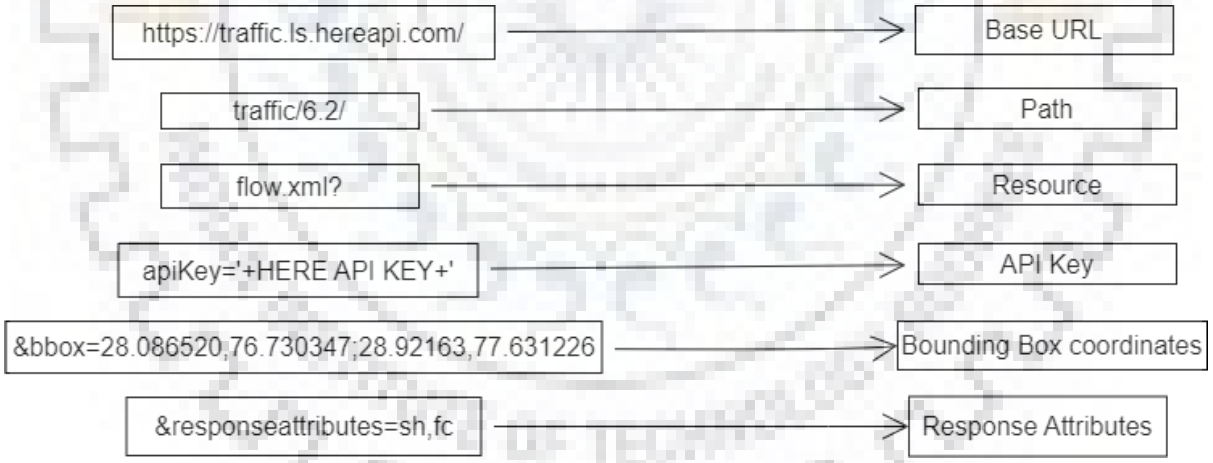

Figure 3.1: Sub Parts of API request

The following data is extracted from the response obtained through API as shown in Fig. 3.2:

- "RWS" : It represents a list of roadway items (RW)
- "RW" : This is the overall flow item for a roadway. Each roadway will have a roadway item with accessible traffic flow information
- "FIS" : It represents a list of flow item elements (FI)

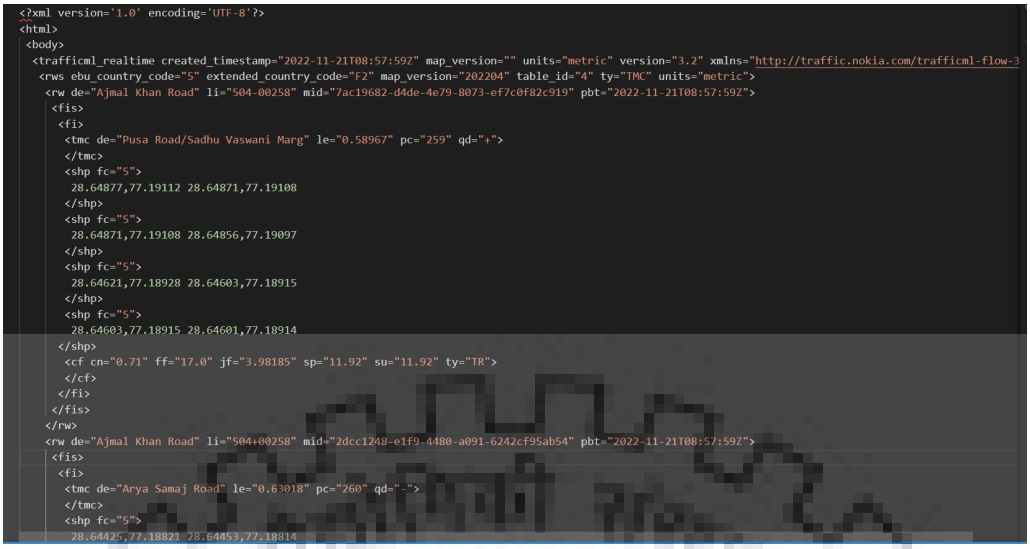

Figure 3.2: XML file from Traffic API

- "FI" : FI represents a single flow item
- "TMC" : TMC stands for "Traffic message channel". Information about location in coded format can be sent and received via TMC if the location code table is integrated with the maps service provided.
- "DE" : It represents the text description of the road
- "CN" : CN stands for Confidence Number indicating percentage of real time data used. Data is said to be in real time if the CN value is greater than 0.7. A value greater than 0.5 and less than or equal to 0.7 indicates historical speeds
- "FF" : FF represents the free flow speed on the given stretch of the road
- "JF" : JF stands for Jam Factor which represents the quality of travel. JF value ranges from 0 to 10 with 10 being a condition of road closure. As the number increases the quality of the travel will degrade
- "SP" : SP denotes the average speed for the road segment. If the speed is above speed limit, then it is capped to the speed limit. The speeds above the speed limit are not taken into consideration.
- "SU" : SU stands for Speed Uncut which also represents average speed. It ignores the speed limit of the road segment

3.2.4 Landuse data

According to the context of our research, landuse data refers to information about how land is utilised, such as for residential, commercial, industrial, or agricultural purposes.

	Names	cn	ff	jf	sp	su	start_point	end_point
2022-11-21 08:51:59	Guru Golnalkar Marg	0.71	18.0	2.71828	6.72	6.72	28.72332,77.06223	28.72323,77.06221
2022-11-21 08:51:59	Rohini Central Market Road	0.91	27.0	5.60884	15.86	15.86	28.72323,77.06221	28.71634,77.05627
2022-11-21 08:51:59	Rohini Central Market Road	0.75	22.0	3.81396	15.71	15.71	28.71642,77.05615	28.71662,77.05634
2022-11-21 08:51:59	Guru Golnalkar Marg	0.83	31.0	4.52766	20.55	20.55	28.71662,77.05634	28.72332,77.06223
2022-11-21 08:51:59	Karawal Nagar Road/Durga Mandir Road	0.71	15.0	0.00000	12.00	12.00	28.73013,77.2723	28.73031,77.27245
2022-11-21 08:51:59	33 Futa Road	0.73	15.0	0.77756	11.13	11.13	28.73031,77.27245	28.73371,77.27535
2022-11-21 08:51:59	Lalbagh Colony Road (South)	0.73	14.0	2.31309	8.57	8.57	28.73371,77.27535	28.73576,77.27748
2022-11-21 08:51:59	Lalbagh Colony Road (North)	0.72	15.0	1.61547	13.18	13.18	28.73576,77.27748	28.74039,77.28144
2022-11-21 08:51:59	Ghaziabad	0.73	18.0	3.79933	12.87	12.87	28.74039,77.28144	28.74221,77.28269
2022-11-21 08:51:59	Loni Road	0.89	18.0	3.70584	13.00	13.00	28.74221,77.28269	28.7436,77.28787

Figure 3.3: Traffic Data extracted from API

This data is essential for various purposes, including urban planning, resource management, and environmental monitoring. Landuse data is extracted from OpenStreetMap (OSM). OpenStreetMap is an open-source platform that provides a wealth of geospatial data, including information about landuse. OSM can be a valuable resource for extracting landuse data, particularly for areas where official data may not be readily available or may need to be updated. In OSM, landuse data is available in shape file (.shp) format. A shape file is a file which stores the geometry of landuse data as polygons in the file.The landuse data is divided into different files such as Point of Interest, buildings, water etc. After processing these files, the categories formed are :

1. Commercial
2. Educational
3. Green
4. Water
5. Industrial
6. Residential

There are various sub categories under each category. The sub categories for each category is shown in Table 3.1. 'Industrial' and 'Residential' categories don't have any sub category.

landuse data is plotted on QGIS and the visualization is shown in Fig. 3.4. Plotting is done for a buffer of radius 8 km for better visualization.

Commercial		Educational	Green		Water
Arts centre	Cinema hall	College	park	farmland	riverbank
Public building	Sports centre	School	track	Nature reserve	wetland
Market Place	Hotels	University	playground	Orchard	water
Museum	Memorial	Hostel	forest	Recreation ground	reservoir

Table 3.1: Sub categories under categories formed for landuse

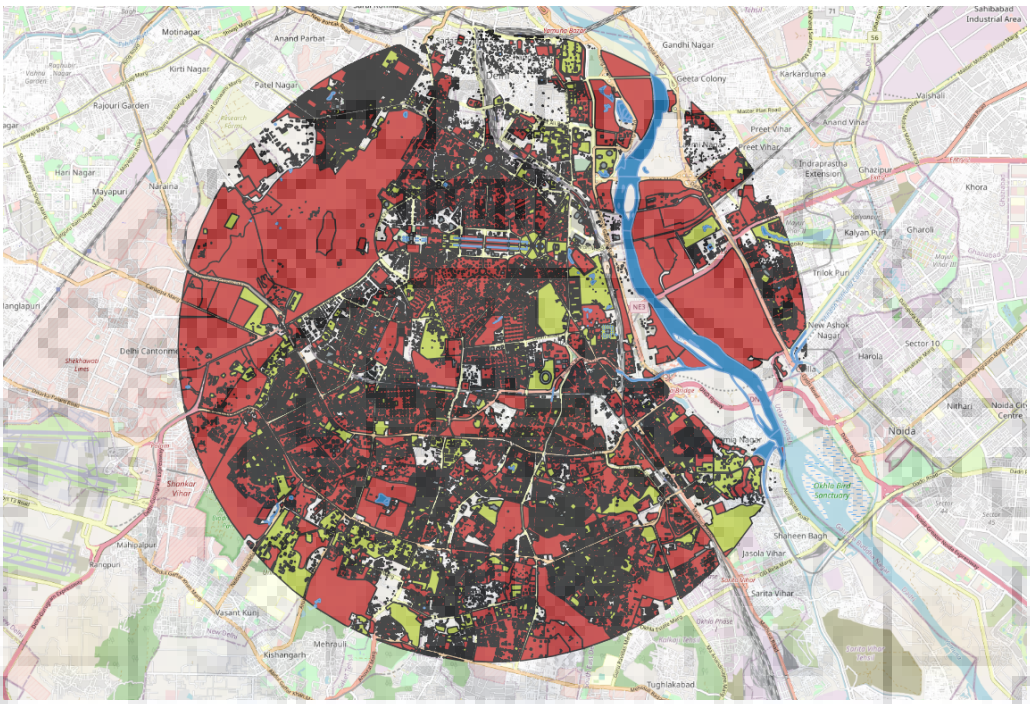

Figure 3.4: Visualization of landuse data in QGIS

3.2.5 Realtime data

Pollution measuring devices were installed in buses in Delhi to get realtime data every 10 minutes. Monitors provide parameters such as $PM_{2.5}$, PM10, Air Quality Index (AQI), Temperature, Humidity, device name and (Lat, Long) of the device. Out of these parameters, $PM_{2.5}$ and lat, long of devices are taken at an interval of 15 minutes. The dataset is taken for six months in the time period ranging from 1 November 2022 to 30 April 2023. Extracted realtime data is shown in Fig. 3.5. There are a total number of 21 different buses with monitoring devices in the extracted dataset.

3.3 Data preprocessing and preparation

We have collected different datasets according to factors affecting air pollution at any location. We have temporal factors such as Atmospheric temperature, Barometric pressure, Relative Humidity, Wind speed and wind direction. landuse data is taken for spatial

	From Date	devicename	lat	long	PM2.5 mobile
0	2022-11-01 00:00:00	Outdoor 17	28.579539	77.229461	332
1	2022-11-01 00:00:00	Outdoor 14	28.579243	77.230473	357
2	2022-11-01 00:15:00	Outdoor 17	28.579426	77.229195	345
3	2022-11-01 00:15:00	Outdoor 14	28.579276	77.230620	348
4	2022-11-01 00:30:00	Outdoor 17	28.579673	77.229560	346
5	2022-11-01 00:30:00	Outdoor 14	28.579463	77.230568	327
6	2022-11-01 00:45:00	Outdoor 14	28.579368	77.230570	318
7	2022-11-01 00:45:00	Outdoor 17	28.579458	77.229400	301
8	2022-11-01 01:00:00	Outdoor 17	28.579378	77.232489	330
9	2022-11-01 01:15:00	Outdoor 17	28.579336	77.231706	312

Figure 3.5: Sample realtime data

variability. The issue is that the datasets are in raw format and must be processed before being used for prediction. Some of the issues are:

3.3.1 Removing overlapping areas from landuse dataset

As stated above, landuse data is extracted from OSM in shape file format. There are three shape files named POI, landuse and buildings. The problem is that the landuse patterns or polygons present in POIs are also buildings but with different classes or names. For example, there is one university, and we have many departments in that university. So the area of the whole university, including departments, is present in the POI file as a different entity, and the area of departments is present in the buildings file as a different category. So this is the overlap in the area when we concatenate two files to find all landuse types in a particular buffer. Also, there are overlaps of areas present in the same shape files. For example, in the POI file, we have a park polygon with fclass as a park. Also, there is a memorial present in the park with fclass as a memorial in the shapefile. So, the area of the memorial is added two times whenever we take the landuse area inside a buffer. To remove this overlap, we had formed two algorithms. One algorithm for removing the overlapping areas from 2 shape files and another algorithm to remove the overlapping areas from the same shape file. Below are two algorithms for removing overlaps from landuse data.

Algorithm 1 is used to remove overlaps from 2 different shapefiles. We take two shapefiles and then check the intersection between the two files. After we have the

Algorithm 1 Remove Overlapping Features from Two Shapefiles

```
procedure REMOVE OVERLAP(file1, file2)
    Intersect file1 and file2 to get intersection
    for all rows in intersection do
        Match osmid1 and osmid2 with corresponding osmids in file1 and file2
        Find area1 and area2 of the polygons from the osmids
        Match fclass of both polygons with their categories
        if category1 != category2 then
            if area1 >= area2 then
                Remove area2 from file1
            else
                Remove area1 from file2
        else
            if area1 >= area2 then
                Remove area2 from file2
            else
                Remove area1 from file1

    return file1, file2 with non-overlapping features
```

intersection of both files, we loop through each of the rows. For each row, find the corresponding osm ids in their respective files and check which of the polygons has a larger area than the others. If the fclass of both the polygons is the same, then the smaller polygon is removed from the respective file and the larger polygon is kept as it is. Otherwise, if the fclass is different for both polygons, then the polygon with a smaller area is removed from the larger one. This is how the overlap is removed from 2 different files. Visualization of Algorithm 1 on a single polygon is shown in Fig. 3.6. In Fig. 3.6, the left side plot shows two overlapping polygons. The blue colour polygon represents the area of a residential building, and the red colour represents a park on the building premises. We can see that the area of the park is coming twice, one in the residential polygon and the other in the park polygon. As the fclass of both the polygons are different, the smaller area that is the park, depicted by red, is removed from the residential polygon that is depicted by blue. Two plots on the right side depicts how the overlap is removed from 2 polygons.

In Fig. 3.7, we can see that the fclass is the same for both polygons. In processing landuse data, we generated categories such as commercial, industrial, green, water, residential etc. The fclass park and grass both come under the green category. So, as they belong to the same category, we have to take the union of both the polygons and the polygon with the larger area is kept in the shapefile.

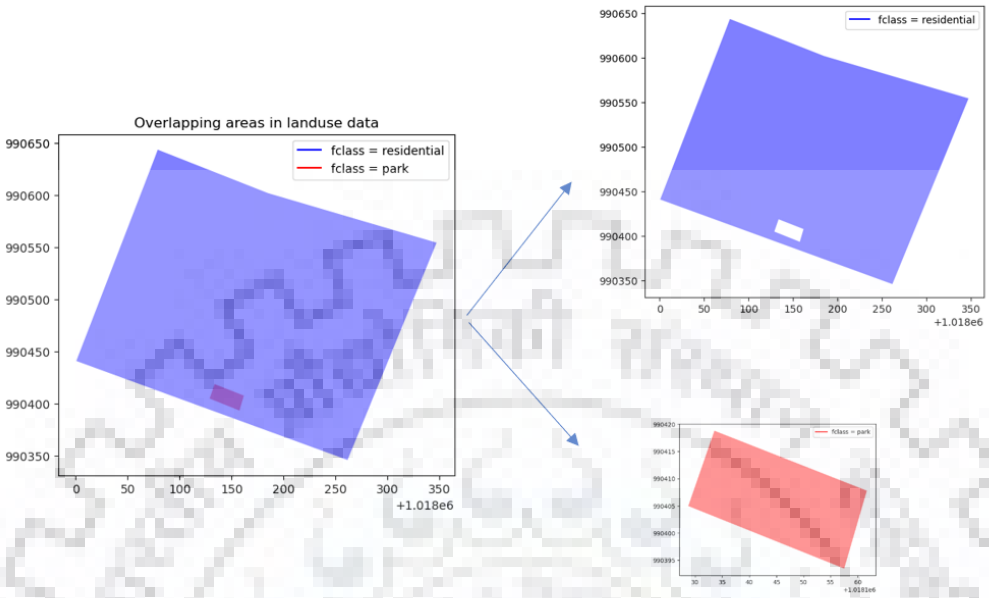

Figure 3.6: Removing overlap from landuse data where polygons belongs to different categories

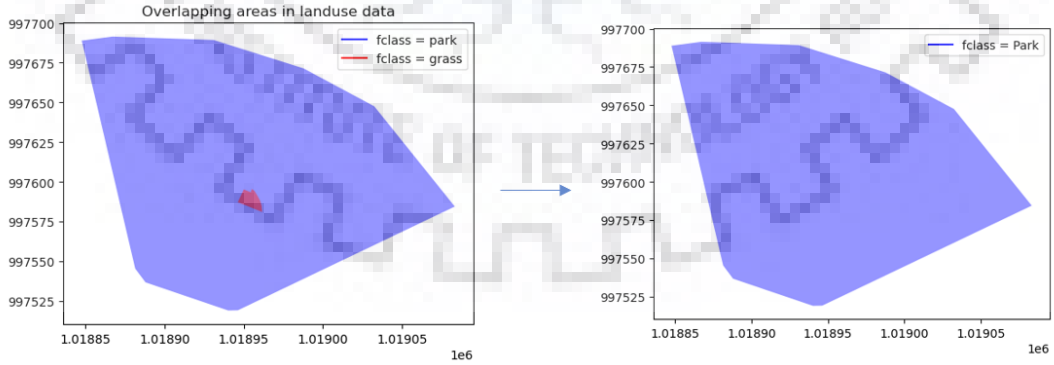

Figure 3.7: Removing overlap from landuse data where polygons belongs to same categories

3.3.2 Data preparation

The objective of our research is to predict the spatiotemporal distribution of $PM_{2.5}$ in areas where no monitoring stations are present. To achieve this, we utilised various datasets, including $PM_{2.5}$ data from static and mobile monitors, landuse data, traffic flow data, and meteorological data collected from the CPCB website. Our methodology began by creating a vector or line string between each static data point and its corresponding mobile data point. We then generated rectangular buffers with varying breadths of 50, 100, 150, and 200 metres based on the length of the vector. We identified the landuse categories that fall within each buffer, which was done by accounting for overlapping data in the landuse dataset. After this step, using the traffic flow data, we determined the lengths of roads with low congestion (congestion factor less than 0.5) and high congestion (congestion factor greater than 0.5) in each buffer. Once we obtained each buffer's landuse and traffic flow data, we merged it with the meteorological data to form the final dataset. It is worth noting that we limited our analysis to five static stations, namely Lodhi Road, Delhi IMD, Jawaharlal Nehru Stadium, Delhi DPCC, Sri Aurobindo Marg Delhi DPCC, Major Dhyan Chand National Stadium, Delhi DPCC, and Nehru Nagar, Delhi DPCC, due to their higher number of real-time data points in the surrounding 1.5 km radius, resulting in a total of 10,107 data points for the six months. This subsection outlines how we synthesised the various datasets to form the final.

$PM_{2.5}$(static) and $PM_{2.5}$(mobile) data

Our research aims to match the $PM_{2.5}$(static) data with the $PM_{2.5}$(mobile) data to create a unified dataset. We used the DateTime and latitude/longitude information available for both datasets to accomplish this. We began by identifying the mobile data points located within a 1.5-kilometre radius of each static point. We used the haversine distance metric to calculate the distance between two points, which provides a distance value in kilometres. The Haversine distance gives the shortest distance between two points on the earth's surface (Wikipedia, 2011) . The distance is calculated using the longitudes and latitudes of the two places. The distance between two stations i and j for k^{th} pollutant is defined in Eq. (3.1)

$$D_{(i,j)} = 2\arcsin\left(\left(\sqrt{\sin^2\left(\frac{(\lambda_j-\lambda_i)}{2}\right)+\cos(\lambda_i)\cos(\lambda_j)\sin^2\left(\frac{\phi_j-\phi_i}{2}\right)}\right)\right) \tag{3.1}$$

Where λ_i,λ_j are latitudes and ϕ_i, ϕ_j are longitudes of the two places i and j respectively.

After finding all the points within the specified radius, we concatenated the resulting five data frames. However, due to the overlap in mobile data points between the five stations, we encountered duplicate entries. To address this issue, we used the *drop_duplicates* function in pandas and provided a subset of columns to identify duplicate rows. Specifically, we used the From Date, device name, latitude(mobile), longitude(mobile), and $PM_{2.5}$(mobile) columns to identify and remove duplicates. The device name column identifies the device to which the portable data point belongs. Thus, any data points with the same device name, location, and $PM_{2.5}$ value simultaneously were considered duplicates and removed from the dataset. With this approach, we matched the $PM_{2.5}$ static and mobile values based on the From Date column.

Landuse data

After, we had matched $PM_{2.5}$ static and mobile data, now we have to form buffers for each row and find the landuse patterns present in each of the buffers. The landuse data extracted from OSM has EPSG:4326, which is latitude, and longitude are in degrees. We have to change CRS in metres to form the buffer in metres.

First of all, we will understand about EPSG and CRS: EPSG (European Petroleum Survey Group) is an organisation that maintains a database of coordinate reference systems (CRS) for geospatial data. CRS is a system that defines how geographic coordinates, such as latitude and longitude, are referenced and displayed on a map. In a geodataframe, EPSG refers to the CRS's EPSG code in which the data is. The EPSG code is a unique identifier for a CRS, and it specifies the projection, datum, and units used in the coordinate system. The EPSG code can convert between different CRSs and project data onto different map projections. CRS, on the other hand, is a set of rules that define how the coordinate system is defined and how the positions on the Earth's surface can be described. It describes how the Earth's surface is approximated and how coordinates are related to this surface. CRS is essential in geospatial data analysis because it allows us to accurately overlay and analyse data from different sources using different coordinate systems.

We have changed the CRS of landuse data from EPSG:4326 to EPSG:7760. EPSG:7760 refers to the coordinate reference system in metres. Also, we had to convert the latitude and longitude of both static and mobile data points to EPSG:7760. Now we run a loop through each of the rows. For each row, a linestring is made between the latitude and longitude of static and mobile data points. After that, a rectangular buffer is created, and the landuse data is intersected with this buffer. This generates an intersected geo-

dataframe containing all the landuse categories in the buffer. After that area of each of the polygons in the intersected geodataframe is found grouping by different categories, and the area of each category is stored in a dictionary. After iterating all rows, each category's areas for each row are stored in the dictionary, and a new dataframe is formed from that dictionary. Now the new dataframe is merged with the dataframe containing $PM_{2.5}$ static and mobile data points. In Fig. 3.8, a buffer of width 150 metres is formed between $PM_{2.5}$ static and mobile data points. Fig. 3.8 shows the distribution of landuse data in the buffer. Red color represents the residential area, green color represents the green area and violet color represents the commercial area.

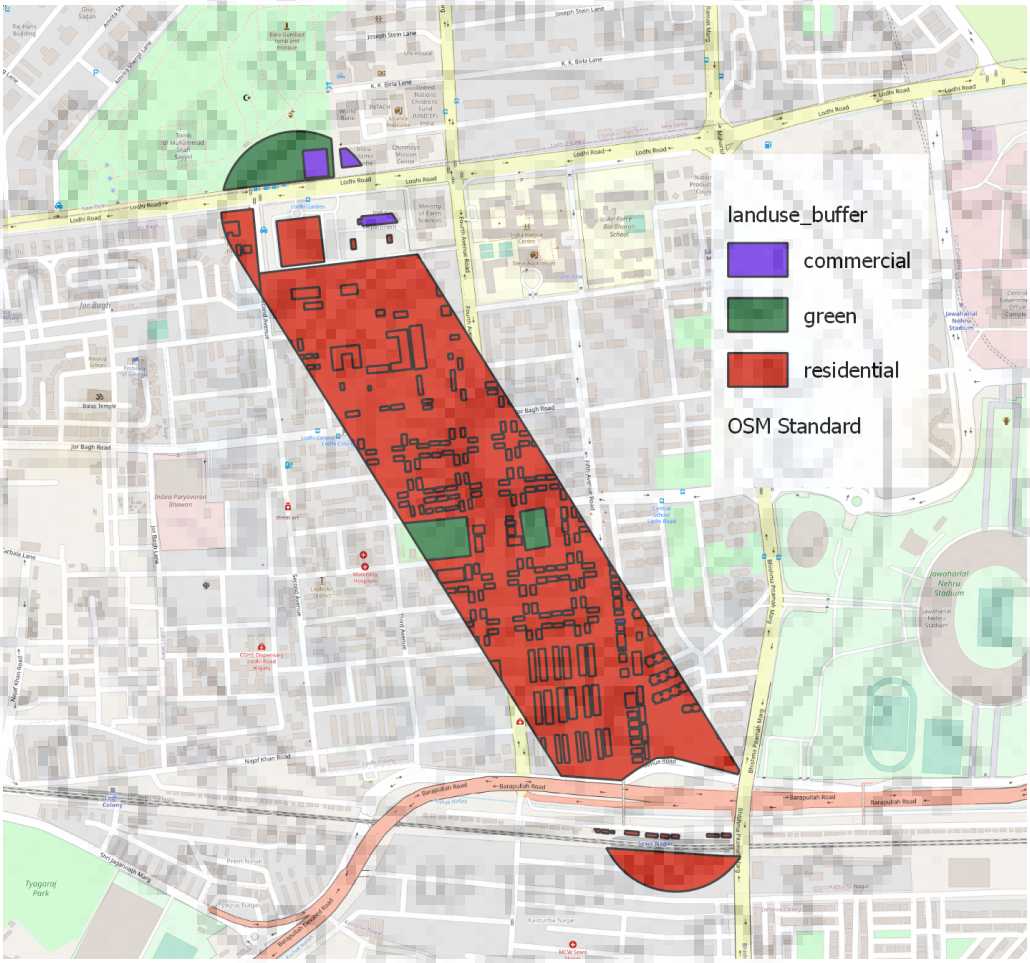

Figure 3.8: Distribution of landuse data in a buffer

Traffic flow data

We used the HERE Maps Traffic flow API to gather and extract traffic flow data every 10 minutes. The extracted data included parameters such as CN, FF, JF, SP, and SU. From these parameters, we derived a new feature called the "congestion factor" using the formula FF/SU, where FF represents the free flow speed of vehicles on a particular road

segment, and SU represents the average speed of vehicles on roads if the speed limit of the roads is ignored. The roads are divided into four categories as shown:

1. $roads\_1 : 0 <= congestionfactor < 0.25$
2. $roads\_2 : 0.25 <= congestionfactor < 0.5$
3. $roads\_3 : 0.5 <= congestionfactor < 0.75$
4. $roads\_4 : 0.75 <= congestionfactor$

The extracted data was stored in Excel files for all roads in Delhi, with each file representing traffic flow data for a specific DateTime. We formed a dataset above, which contains DateTime and respective $PM_{2.5}$ static and mobile data points. For using traffic flow data, we created four categories of roads as roads_1, roads_2, roads_3, and roads_4. For making this dataset, we have to match the DateTime from the dataset created above with the file names extracted from here maps API. We have taken the DateTime from the dataset and then formed a path with the DateTime for the directory where the file is present. Try and except exception is used to handle the cases when the file for particular DateTime does not exist. After finding the csv file with the specific DateTime, we formed the shapefile from this file to find the length of roads within a given buffer. We took latitude and longitude from the above dataset for the static and mobile data points. A point geometry is formed from these latitudes and longitudes and uses these points to create a linestring between these two points. After this, we formed a rectangular buffer of different widths ranging from 50 metres to 200 metres. Now intersection is done between the shapefile of traffic flow data and the buffer geometry. After getting the intersected shapefile, we formed four dataframes for each road category. Then we have found the length of road segments for each road category and stored them in their respective lists. After iterating through the above dataset, we added the traffic data to the above dataset.

DateTime features

For using Machine learning, ensemble models and Artificial Neural Networks for prediction, we have to process the DateTime column for considering the temporal effect of the dataset on the prediction. Hence, we have developed some features from the DateTime column. The features are as follows: 15th minute of the day, day of the week and week of the year.

1. 15th minute of the day: In an hour, if we consider an interval of 15 minutes, then we will have 4 data points each hour. There are 24 hours in a day, so we have a

total of 96 data points each day. So, the 15th minute of day represents that the dataset belongs to which 15th minute of a particular day.

2. Day of the week: In a week there are 7 days. This feature represents that this day is which one out of 7 days in a week.
3. Week of the year: It tells us the week for which we are finding the day and the 15th minute of that day is which week of the year. In a year there are 52 weeks for a non leap year. But we have taken data for 6 months and 10 days. Hence we have only 28 weeks in our data.

These 3 features are used to correctly identify the temporal pattern in the dataset. It is used to check if there is any seasonality in the dataset. These 3 features have datatype as 'object'. So we need to convert these features into numerical data type using encoding techniques. Most commonly used encoding techniques are: One hot encoding and label encoding.

Label encoding

Any variable with two or more categories (values) is referred to as a categorical or discrete variable. Nominal and ordinal categorical variables are two different types. A nominal variable's categories do not naturally have an inherent order. For instance, gender is a categorical variable that has two categories (male and female), neither of which has any inherent ordering. The ordering of an ordinal variable is obvious. Take pollution as an example, which has three categories: low, medium, and high. Label encoding is used for ordinal categorical variables. Using the Label Encoding technique, categorical columns can be transformed into numerical ones that can be fitted by machine learning models that only handle numerical data. In a machine-learning project, it is a crucial pre-processing phase. Label encoding can be applied only to ordinal categorical variables. Ordinal categorical variables are the categories which have an inherent order. Suppose we have a column traffic flow in a dataset that has elements as Low, Medium, and High. We can apply label encoding to this column as it is an ordinal categorical variable. As shown in Table 3.2 that after applying label encoding, the traffic flow column is converted into a numerical column with values as 0,1 and 2, where 0 is the label for low, 1 for medium and 2 for high traffic flow.

Traffic Flow	Traffic_flow_encoded
Low	0
Low	0
High	2
Medium	1

Table 3.2: Label Encoding

One hot encoding

One hot encoding is used for nominal categorical variables. It is used to encode categorical datasets into numerical data that will be appropriate for Machine learning models. It creates a binary feature for each unique category in the categorical feature. For example if there are three categories in the categorical column, then three rows will be made, each representing a category. After that, the value in the column which belongs to a category becomes 1, and all other column values become 0 for a particular row. Suppose we have a column with 3 colors that are Red, Green and yellow, as shown in Table 3.3. The first column in the table represents categorical variables, and the next three columns represent one hot encoded data.

Color	Green_encoded	Yellow_encoded	Red_encoded
Green	1	0	0
Green	1	0	0
Yellow	0	1	0
Red	0	0	1
Red	0	0	1

Table 3.3: One Hot encoding of nominal categorical variables

Hence, we will use one hot encoding for the DateTime features that we have created above. After applying one hot encoding, the number of features increases and below are the number of features obtained:

1. 15th minute of the day : 96 columns
2. Day of the week : 7 columns
3. Week of the year : 28 columns

In total, we have 131 columns that are formed from the Datetime feature.

3.4 Models and their architecture

3.4.1 Ensemble models

Machine learning models use a single model for prediction tasks. Anytime we want to make any decision in life, we tend to collect as much data as possible and take advice from many people. The more information, the more the probability of our decision being correct. Similarly, machine learning is a mathematical model that observes the data's patterns and dependencies and predicts the output based on the patterns observed. The more data, the more accuracy of predictions. But in most cases, more than a single model is required for accurate predictions. This drawback of machine learning models is handled by ensemble methods. Ensemble learning methods also use machine learning models, but instead of a single model, they use multiple models for the prediction. These models are also called base estimators. The disadvantage of using a single estimator is:

1. High Variance: A single model becomes very sensitive to changes in the patterns in the data such that it considers noise and outliers as a pattern.
2. High bias: The model becomes so much less sensitive to changes in the patterns that it ignores the noise and the basic patterns in the dataset.

Different types of ensemble learning methods are Bagging and Boosting.

Bagging: Bagging is a parallel learning algorithm. Multiple learners are used in bagging. Multiple subsets of data are created from the dataset with repetition. For example, we have a dataset with 100 rows, and we want to make a bagging model of 5 estimators, then we have to make 5 subsets of data from the given dataset. Suppose we make each subset of size 20. We will make the first subset by selecting random data from the dataset. After that, the subset is not removed from the original dataset. It means that the rows in one subset can also occur in another. It helps in reducing variance and hence reducing the overfitting of the model. Random Forest is a type of bagging algorithm. Multiple parallel decision trees are taken in Random Forest, and the data subsets are given to each decision tree. For classification, we take the category with the highest number of repeating instances. In regression, we can take the mean of all the predictions coming from each decision tree. Fig. 3.9 shows the working of bagging algorithm. In random forest, models were taken as Decision trees. We do majority voting for classification tasks and averaging for regression tasks.

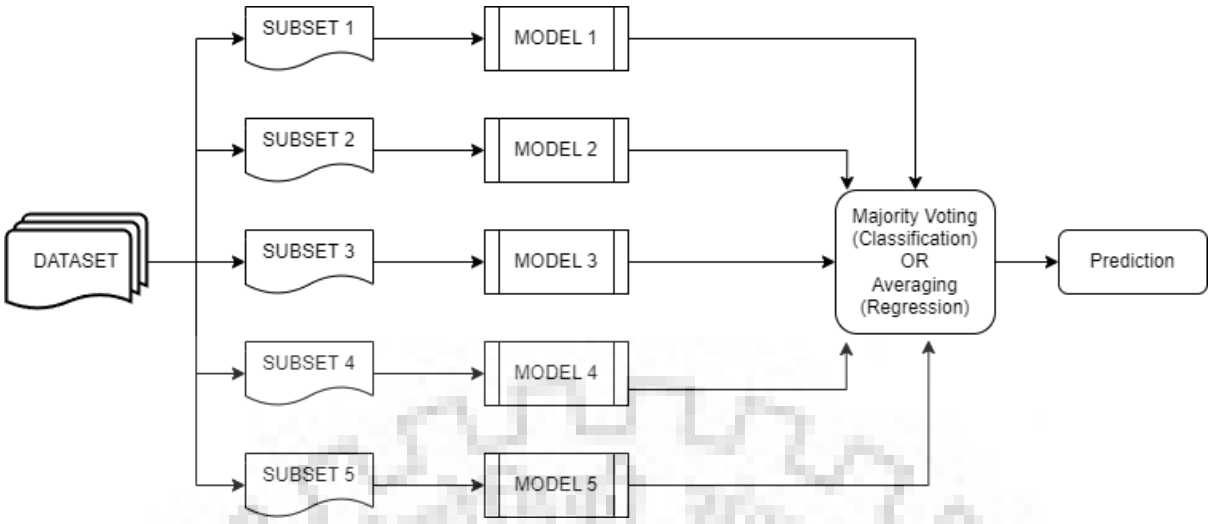

Figure 3.9: Bagging algorithm

Boosting: Boosting: Boosting algorithm uses weak learners for the prediction task. Boosting training is done sequentially compared to the parallel manner done in the bagging algorithm. The process of training continues till the error reduces below a threshold. Different types of boosting algorithms are:

1. Adaptive Boosting (AdaBoost): This algorithm is used for classification tasks. For every iteration, Adaboost identifies the points that are misclassified and increase their weights so that the next learner gives more attention to the misclassified points. It also reduces the weights of correctly classified points.
2. Gradient Boosting: Gradient boosting algorithms are used for both classification and regression tasks. Some of the popular gradient-boosting algorithms are LightGBM and XGBoost. Gradient boosting does not focus on changing the weights of misclassified points. Instead, it focuses on reducing the difference between the predicted and original values. It can easily handle datasets with large dimensions without overfitting in the model. These type of algorithms can easily identify the non linear relationship between the dependent and independent variables.

3.4.2 Artificial neural networks

Artificial Neural Networks (ANN) is a machine learning algorithm inspired by how the human brain processes information. It can identify complex patterns and relationships in the dataset. ANN works with the dataset where conventional machine learning models may suffer to work. They composed of interconnected nodes called neurons, arranged in layers that process and transmit data. The ANN structure typically comprises three

layers: the input, hidden layer(s), and output layers. The input layer receives the input data, which is then processed by the neurons in the hidden layer(s), and final, the output layer produces the network's final output. The complexity of the problem being solved determines the number of hidden layers and neurons in each layer. ANN employs weights and biases that assign values to each neuron, influencing it impact on the network's output. During training, these values adjust to minimize the difference between the network's output and the expected result. Backpropagation is the most commonly used training algorithm for ANN. It involves propagating the error from the output layer back through the network, allowing adjusting the neurons' weights and biases. To summarize, ANN is a network of interconnected neurons arranged in layers. The network trained using algorithms such as backpropagation to adjust weights and biases, minimizing the output error. Understanding the structure and function of ANN is vital to create effective machine-learning models.

Activation functions

Activation functions are used in Artificial Neural Networks (ANNs) to figure out what the neurons should output. There are different activation functions like sigmoid, ReLU, tanh, and Softmax, and they have different advantages and disadvantages.

The sigmoid function is popular activation function. It maps input values to a range between 0 and 1. If the input value to a neuron is greater than 1, then it is clipped to 1 and the neuron's output is 1. Similarly if the input to neuron is less than 0, then it is made 0. It is a monotonic and differentiable function. It is represented as: $f(x) = \frac{1}{1+e^{-x}}$ The difference between sigmoid and softmax function is that sigmoid function is used for binary classification tasks and softmax function is used for multiclass classification tasks. Other activation functions include the hyperbolic tangent function (tanh), which is similar to the sigmoid function but outputs a value between -1 and 1.

ReLU is the most popular and widely used activation function in thw world. It is a simple activation function represented by the formula : $f(x) = \max(0, x)$. We can understand from the formula that when the input to a neuron is positive, then the output of the neuron is same as input and if the input value to the neuron is negative, then ReLU makes it zero. ReLU function is both monotonic and differentiable.

The choice of activation function can have a substantial impact on the performance of an ANN. For the development of effective and efficient ANNs for various machine learning tasks, it is crucial to comprehend the characteristics and uses of activation functions.

Batch normalization

Batch normalization is one of the methods to reduce overfitting in neural networks. It further improves the training speed and stability of neural networks. In batch normalization, the inputs to each layer are normalized, which helps to solve the problem of internal covariate shift. Batch normalization normalises each layer's inputs to have zero mean and unit variance.

For a given mini batch:

1. Find mean of the mini batch

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{3.2}$$

   Here μ_b is the mean of mini batch, x_i is the input value and m is the batch size

2. Find the variance of mini batch

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{3.3}$$

   Here σ_B^2 is the variance of mini batch

3. Normalize the inputs in the mini-batch

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{3.4}$$

   Here $\hat{x}_i$ is the normalized value of i^th batch.

4. Normalized value $\hat{x}_i$ is then transformed using following formula

$$y_i = \gamma\hat{x}_i + \beta \tag{3.5}$$

   Here γ is the scaling parameter and β is the shift parameter. Both are learnable parameters.

3.4.3 Long short term memory

LSTM is a type of Recurrent Neural network (RNN) that can be used in various domains such as Time series forecasting and Natural language processing. LSTMs are a special kind of RNN that can learn long term dependencies. The horizontal line on top in called cell state. LSTMs have the ability to add or remove some information from the cell state. The addition or removal of information is controlled by the gates. Firstly, we have to decide, which information to keep and which to delete. This decision is made by a

sigmoid layer called the forget gate layer. It looks at h_{t-1} and x_t and a number between 0 and 1 is generated for each value in cell state.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3.6}$$

After that, we have to decide which information we are going to store in the cell state. A sigmoid layer called 'input gate layer' decides which values need to be update. A tanh layer creates a vector $\tilde{C}_t$ that could be added to the state. Now, we will update the old cell state C_{t-1} to the new cell state C_t. For this, we multiply old cell state by f_t for forgetting the things. After that we add $i_t * \tilde{C}_t$ to this.

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \\ \tilde{C}_t &= \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \end{aligned} \tag{3.7}$$

Finally, we will decide what should be the output. It will be based on our cell state. Above, we used sigmoid function to decide which information to keep and which one to forget as shown in Eq. (3.6). After that, we pass the cell state through tanh and multiply it by the output of sigmoid gate so that only desired information reaches the output. The output h_t is given by Eq. (3.8)

$$\begin{aligned} o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \\ h_t &= o_t * \tanh\left(C_t\right) \end{aligned} \tag{3.8}$$

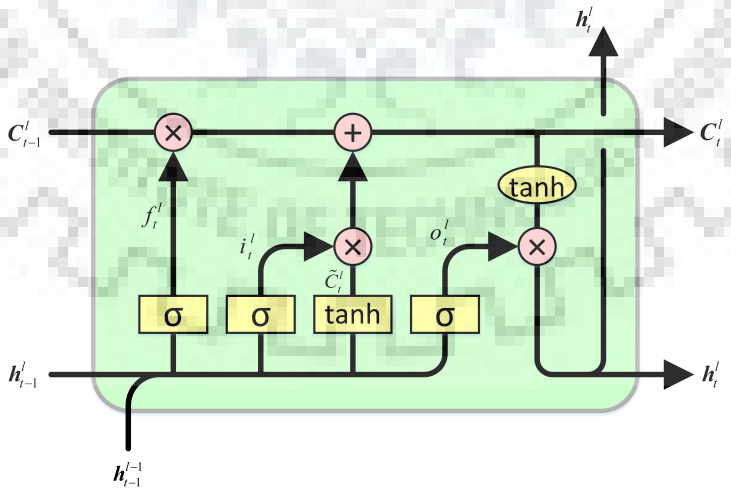

Figure 3.10: LSTM Architecture (StackOverflow, 2018)

Chapter 4

Results and Discussion

4.1 Data analysis

The final dataset is created, and it contains spatial and temporal features. Spatial features are the distance between static and mobile data points and landuse data categories such as green, residential, commercial, industrial, and water, and length of roads according to congestion factor. Temporal features are $PM_{2.5}$ (static), and meteorological features such as Atmospheric Temperature, Barometric pressure, Relative Humidity, and Wind Speed. We have a total of 18336 rows in the dataset. Fig. 4.1 represents histogram plot for $PM_{2.5}$ static and $PM_{2.5}$ mobile. The Histogram of $PM_{2.5}$ static is right-skewed with a skewness of 1.32. Positive skewness means that the tail of the distribution is skewed towards the right, indicating a longer right tail and relatively fewer extreme values on the left side. The mean and median for $PM_{2.5}$ static are 118.122 $\mu g/m^3$ and 105.74 $\mu g/m^3$, respectively. The mean value is slightly more than the median, which indicates that there are some high values in this column. We have removed outliers from our data on a daily basis because if we remove outliers from the whole data, then the days where the pollution level is actually high may be removed. This is the reason our histogram plot is right skewed. $PM_{2.5}$ mobile data is also right-skewed with a skewness of 1.15. Mean and median values for the data are 109.29 $\mu g/m^3$ and 89.66 $\mu g/m^3$. There are more extreme values in the data. This is due to the reason that in $PM_{2.5}$ mobile data, we have data for many stations, which makes it very dynamic. Fig. 4.2 shows a histogram plot for Atmospheric Temperature and Wind Speed. Atmospheric temperature is slightly right skewed with the skewness of 0.350056. Mean and median values are 20.195°C and 20.400°C respectively. It infers that there are not many extreme values in this data. Wind speed is slightly left-skewed with a skewness of -0.827592. Mean and median values are 1.024 m/s and 1.10 m/s, respectively. It infers that the data points are evenly distributed.

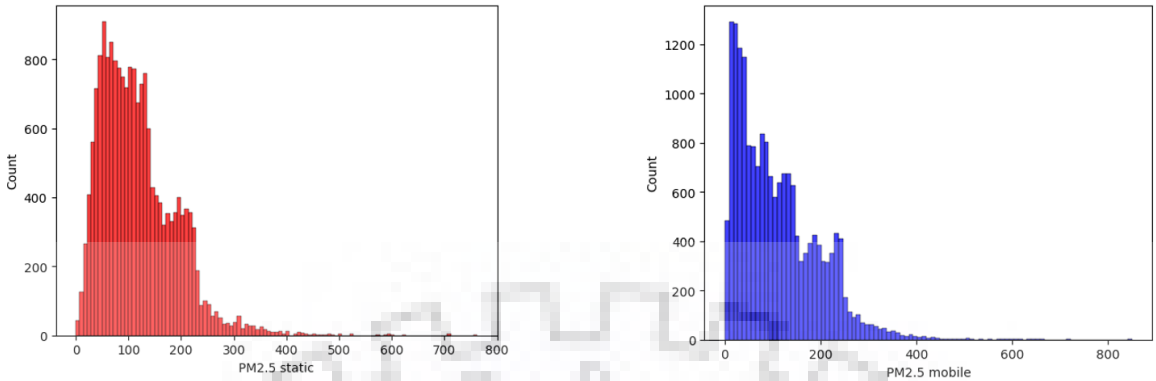

Figure 4.1: Histogram for $PM_{2.5}$ static and $PM_{2.5}$ mobile

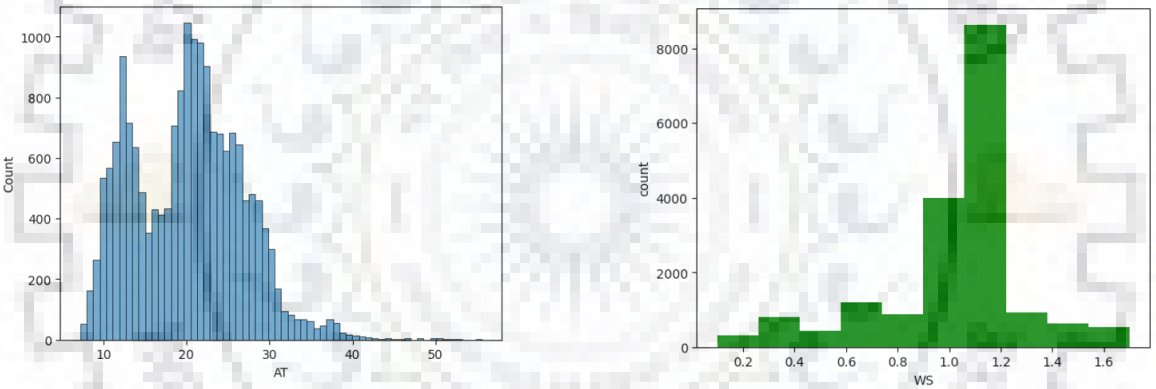

Figure 4.2: Histogram for Atmospheric Temperature and Wind Speed

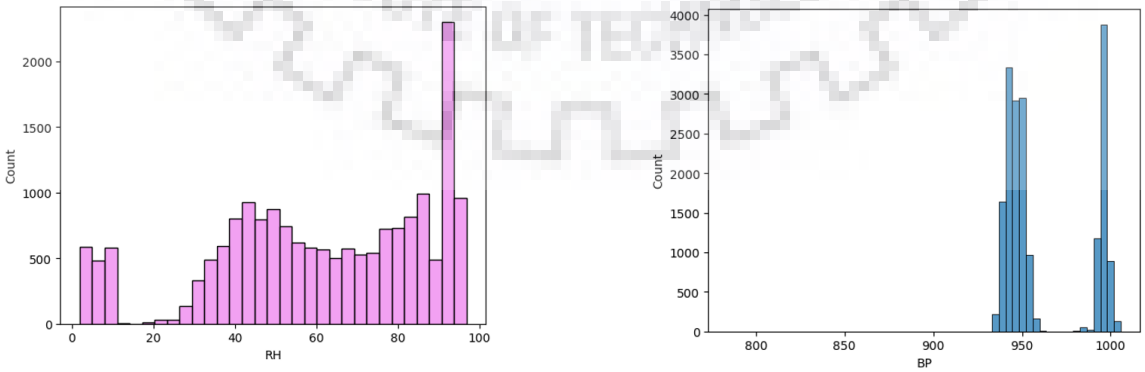

Figure 4.3: Histogram for Relative Humidity and Barometric Pressure

Fig. 4.3 shows a histogram plot for Relative humidity and Barometric pressure. They have a skewness of -0.531401 and 0.591725, respectively. The mean and median for relative humidity are 61.304 % and 63.10 %, respectively. The mean and median for Barometric pressure are 962.619 mbars and 949.700 mbars, respectively.

4.2 Correlation analysis

In our dataset, we have spatial as well as temporal features. Fig. 4.4 shows a plot of the correlation of $PM_{2.5}$ static and mobile data with temporal variables. It can be observed that the predictor variable, that is, $PM_{2.5}$ mobile, is highly positively correlated with $PM_{2.5}$ static. There is a positive correlation between the predictor variable with Barometric Pressure and Relative Humidity. It has a negative correlation with Atmospheric Temperature and Wind Speed. It is inferred from the correlation matrix that particulate matter that is $PM_{2.5}$ increases if Atmospheric temperature and Wind Speed decrease. Higher wind speeds disperse the pollutants. It enhances the mixing and dispersion of air pollutants, reducing their concentration in a specific area. Due to the dilution and dispersion of air pollutants, $PM_{2.5}$ reduces with an increase in wind speeds. $PM_{2.5}$ decreases with an increase in Atmospheric temperature due to the reason that Wind speed increases with an increase in Atmospheric temperature and more dispersion of particulate matter happens. The dataset is taken from 1 November 2022 to 30 April 2023. In this time period, relative humidity is very less, due to which a positive correlation is seen between $PM_{2.5}$ and relative humidity. From the literature, it is known that $PM_{2.5}$ decreases with an increase in humidity.

Fig. 4.5 shows correlation of $PM_{2.5}$ static and mobile data with spatial variables. $PM_{2.5}$ is positively correlated with commercial areas, industrial areas, and the length of roads. In the commercial areas, there will be high particulate matter concentration as these are the areas where more population visits frequently. Due to this, there will be more traffic congestion in these areas compared to areas with residential buildings. In industrial areas, there will be machinery and industrial works that generate pollutants such as $PM_{2.5}$, PM_{10}, and nitrogen oxides. This leads to an increase in the concentration of pollutants and higher pollution. $PM_{2.5}$ is most positively correlated with roads_3 and roads_4 as these categories of roads represent the roads where the congestion factor is greater than 0.5. There will be high emissions of pollutants due to high congestion, which further leads to high air pollution. $PM_{2.5}$ is negatively correlated with green areas, residential areas, and water areas. $PM_{2.5}$ decreases with an increase in green areas which

is due to the fact that leaves and tree canopy provide more surface area for the particulate pollutants to settle as well as they also restore normal gaseous concentration.

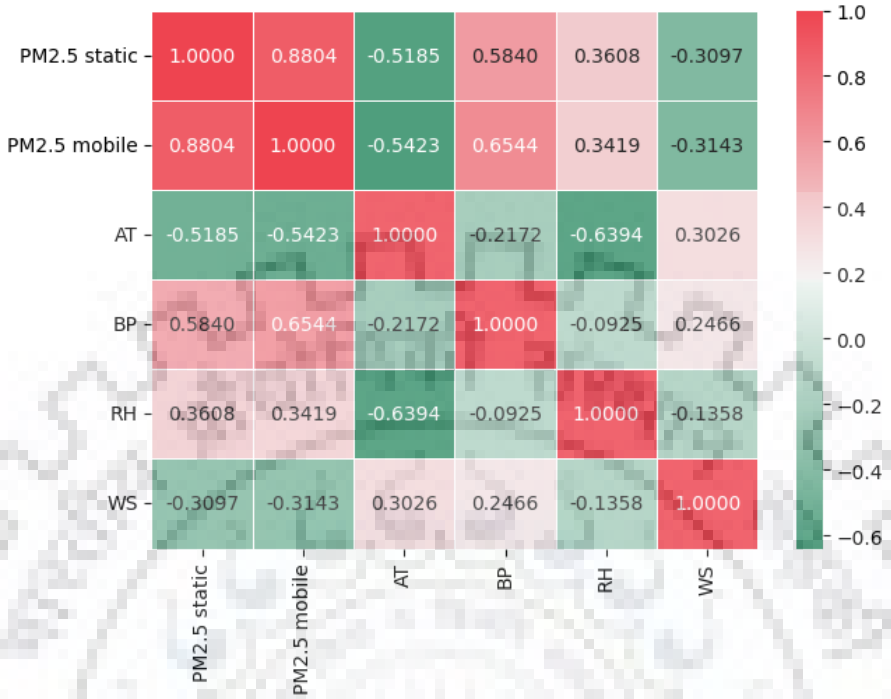

Figure 4.4: Correlation of $PM_{2.5}$ static and $PM_{2.5}$ mobile with temporal variables

4.3 Model development and evaluation

The aim of our project is prediction of $PM_{2.5}$ at locations without monitoring stations. We have spatial as well as temporal variables in our dataset. We have used Artificial neural network for spatial variables and time series models such as Long Short Term memory (LSTM) and Gated Recurrent Units (GRU) for temporal variables.

XG boost

XG Boost model is used as a base machine learning model for comparison with results from deep learning models. The parameters used are max_depth equals to 10, number of estimators as 50 and alpha value as 10. The results obtained from XG Boost are shown in Fig. 4.6

ANN architecture

1. Input layer: There are 147 units in input layer which represents the number of features.

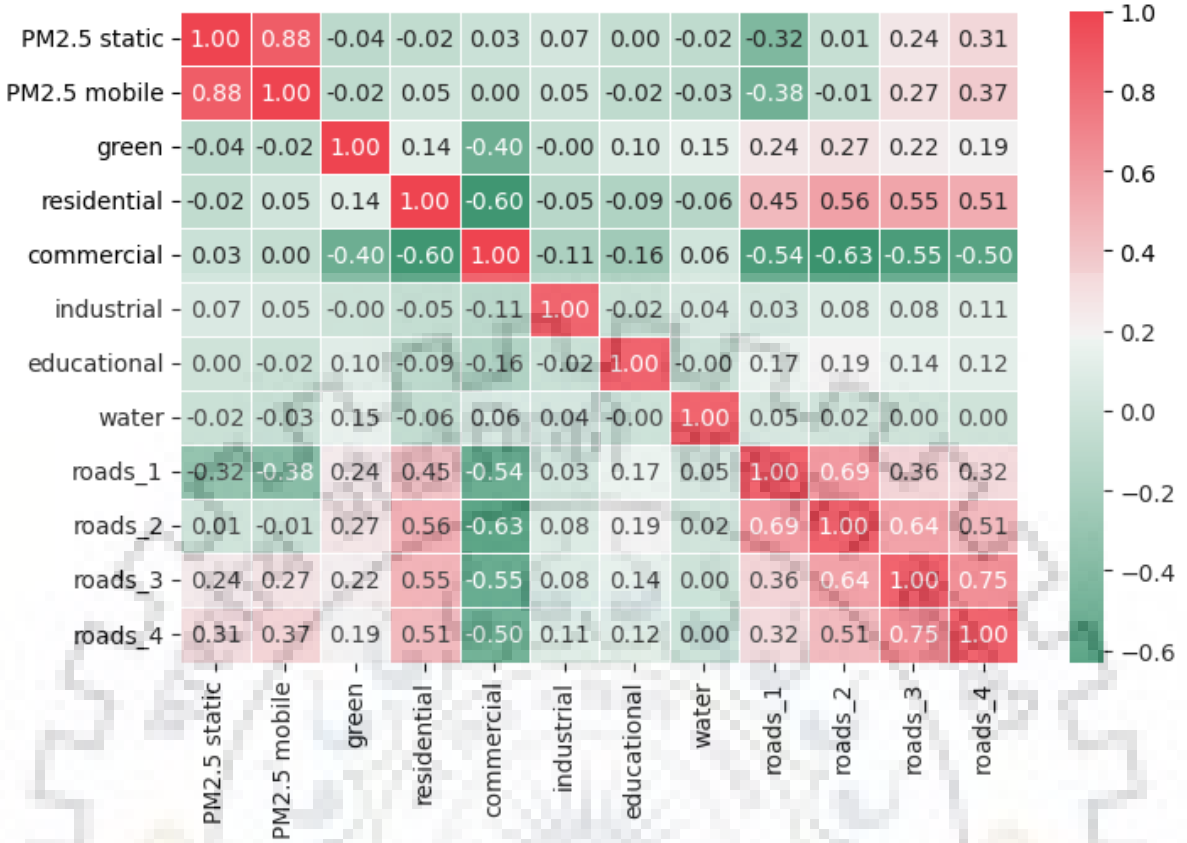

Figure 4.5: Correlation of $PM_{2.5}$ static and $PM_{2.5}$ mobile with spatial variables

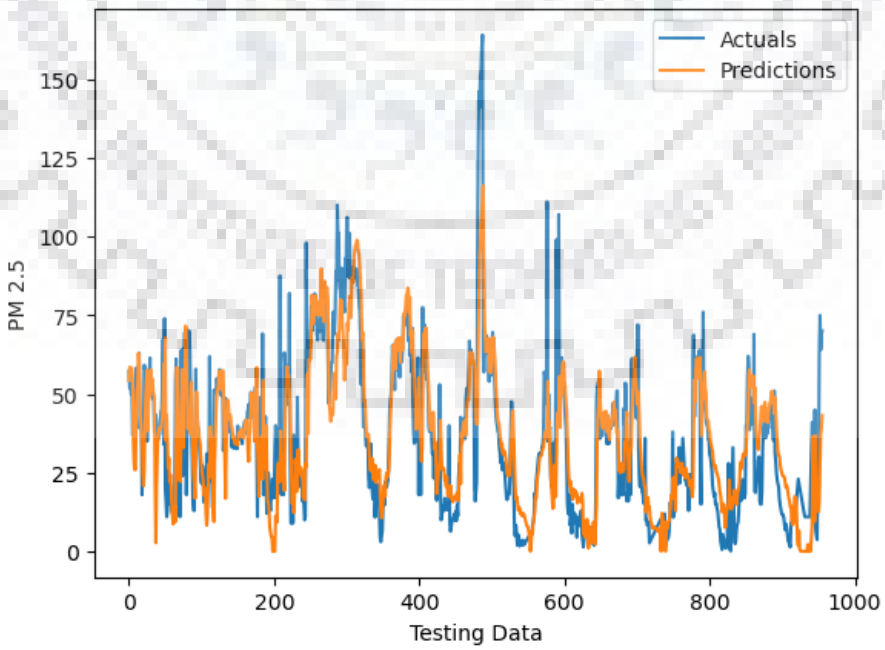

Figure 4.6: Actual and Predicted results on test data using XG Boost

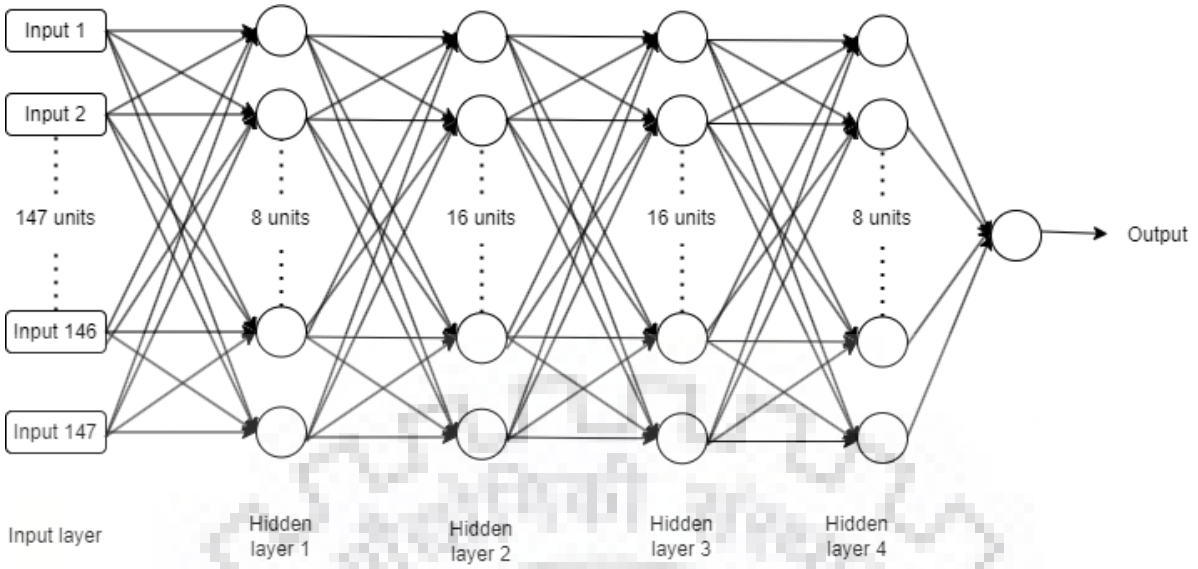

Figure 4.7: ANN architecture for $PM_{2.5}$ prediction

2. Hidden layer 1: There are 8 neurons used in hidden layer 1 followed by batch normalization layer. Relu activation function is used in this layer.
3. Hidden layer 2: There are 16 neurons used in hidden layer 2 followed by batch normalization layer. Relu activation function is used in this layer.
4. Hidden layer 3: There are 16 neurons used in hidden layer 3 followed by batch normalization layer. Relu activation function is used in this layer.
5. Hidden layer 4: There are 8 neurons used in hidden layer 4 followed by batch normalization layer. Relu activation function is used in this layer.
6. Output layer: There is 1 neuron used in output layer. Linear activation function is used in this layer.

LSTM-ANN architecture

Fig. 4.9 shows a hybrid model used for spatiotemporal predictions. The LSTM model is used for temporal predictions, and ANN is used for spatial predictions. In this approach, we have used two layers, each containing 32 units of LSTM. Instead of temporal predictions, we took temporal vector embeddings from the hidden state of the LSTM layer. The LSTM hidden state represents a compressed and abstract representation of the temporal features. The hidden state representation learned by the LSTM model can capture valuable temporal patterns. These vector embeddings are provided as an input to ANN along with spatial features present in the data.

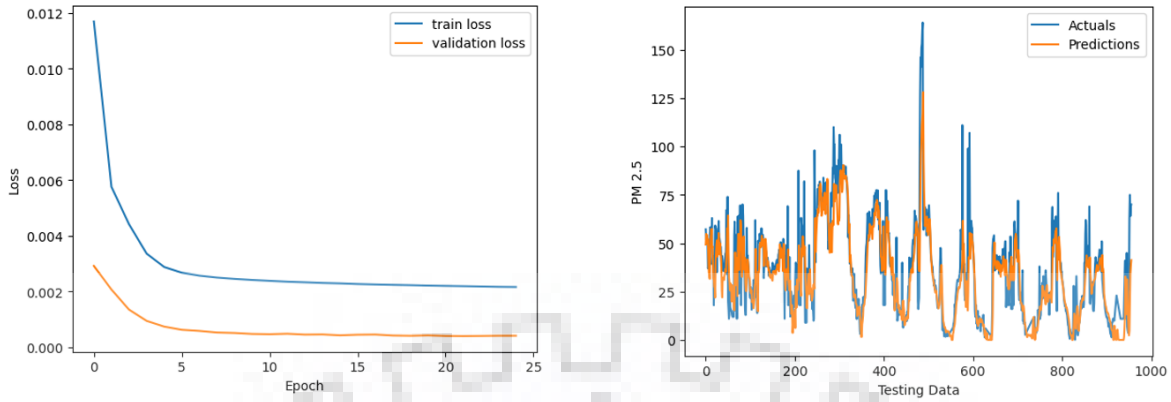

Figure 4.8: Training and Validation loss plot and Plot for actual and predicted values for buffer size of 50 meters using ANN model

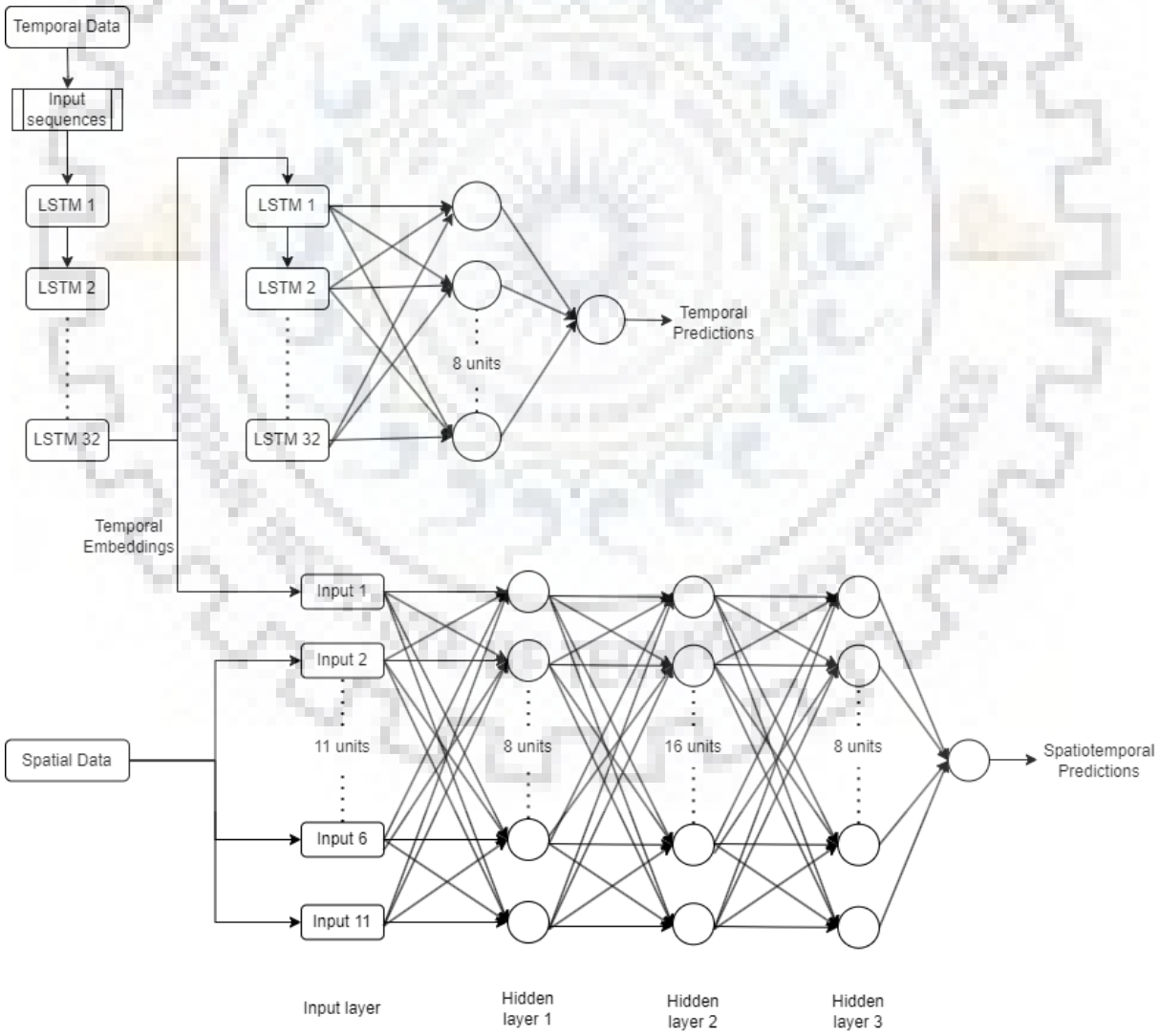

Figure 4.9: LSTM-ANN architecture for $PM_{2.5}$ prediction

In ANN, three hidden layers are used, containing eight, sixteen, and eight neurons, respectively. The relu activation function is used for each of the neurons. The linear activation function is used in the output neuron. Best performance is observed when the buffer size equals 100 meters, and the number of epochs is equal to 50.

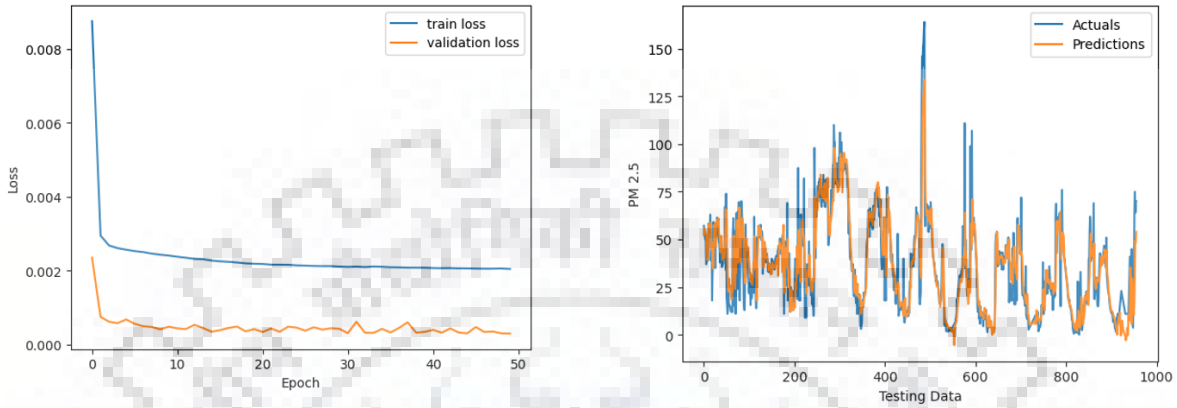

Figure 4.10: Training and Validation loss plot and Plot for actual and predicted values for buffer size of 100 meters using LSTM-ANN model

GRU-ANN architecture

This is a hybrid model that is used for spatiotemporal predictions. GRU model is used for temporal predictions, and ANN is used for spatial predictions. In this approach, we have used two layers, each containing 32 units of GRU. Instead of temporal predictions, we took temporal vector embeddings from the hidden state of the GRU layer. The GRU hidden state represents a compressed and abstract representation of the temporal features. The hidden state representation learned by the GRU model can capture valuable temporal patterns. These vector embeddings are provided as an input to ANN along with spatial features present in the data. GRU has a simple architecture than LSTM. It combines the forget and input gates into a single update gate and merges the cell state and hidden state into a single hidden state.

In ANN, three hidden layers contain eight, sixteen, and eight neurons, respectively. The relu activation function is used for each of the neurons. The linear activation function is used in the output neuron. The best performance is observed when the buffer size equals 50 meters and the number of epochs equals 40.

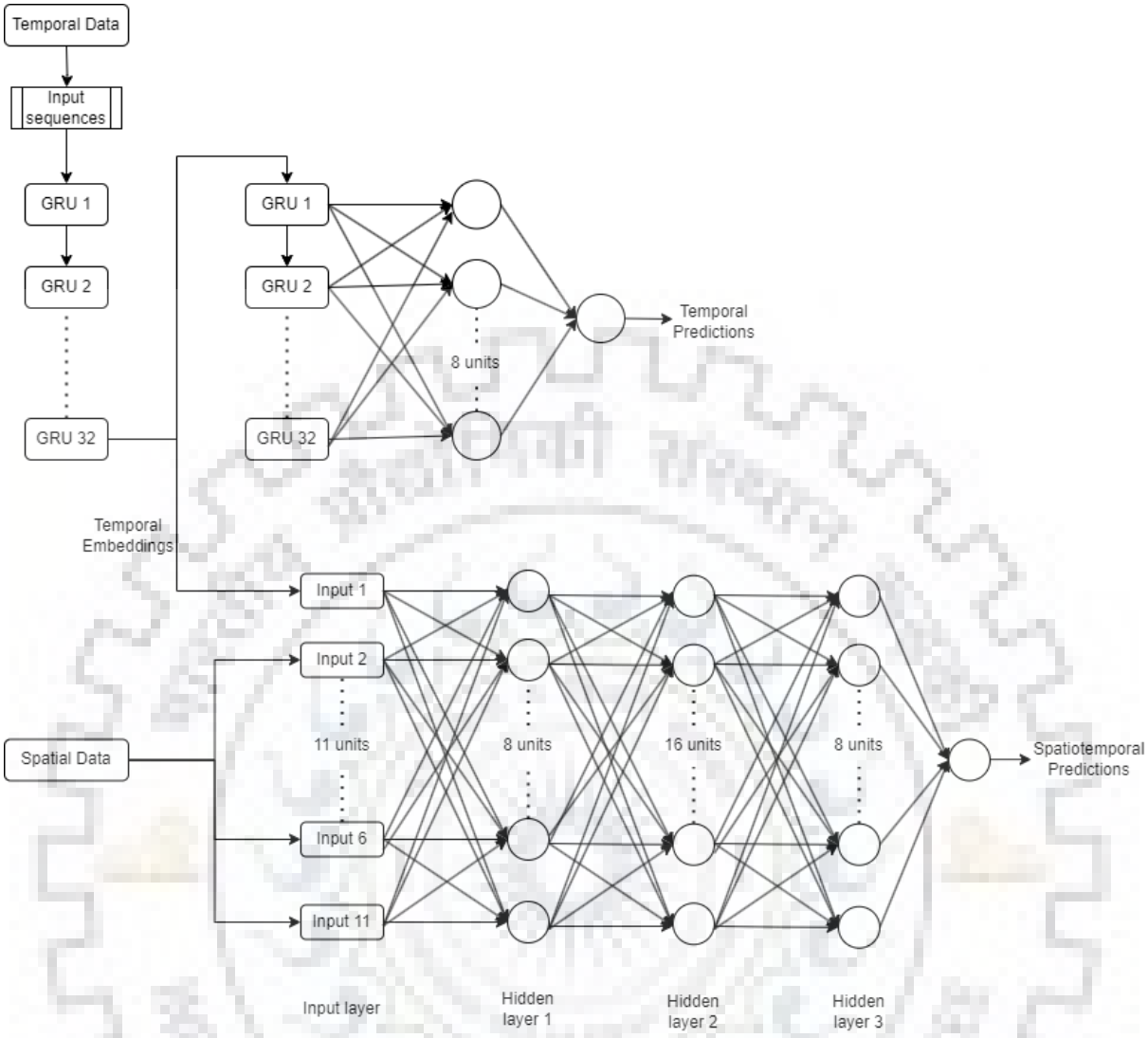

Figure 4.11: GRU-ANN architecture for $PM_{2.5}$ prediction

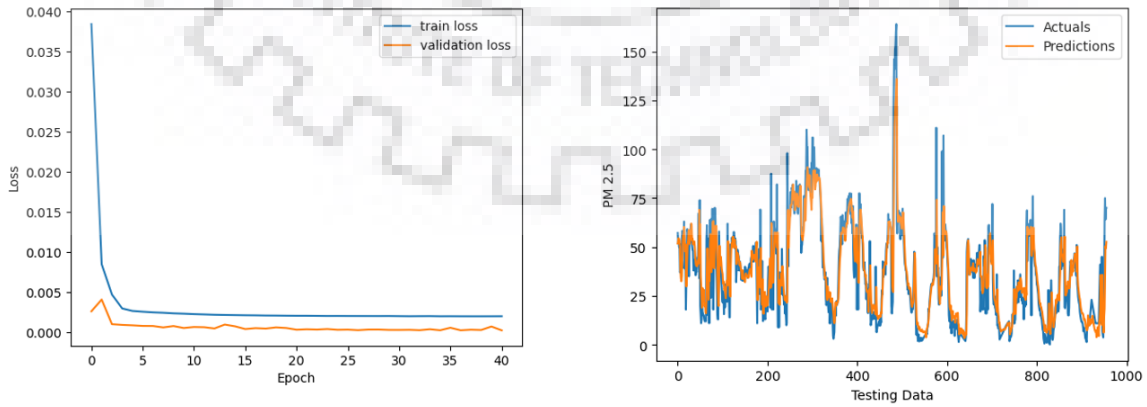

Figure 4.12: Training and Validation loss plot and Plot for actual and predicted values for buffer size of 50 meters using GRU-ANN model

4.3.1 Comparison of results

We have used four different models for spatio-temporal prediction of $PM_{2.5}$ at different locations. These four models are:

1. XG Boost
2. ANN
3. LSTM-ANN
4. GRU-ANN

XG Boost, along with time-dependent features, is used as a base model for comparison of deep learning models. Each model is trained for four buffer sizes that are 50 meters, 100 meters, 150 meters, and 200 meters. The best results for XG Boost are obtained at the buffer of the breadth of 50 meters.

ANN model is trained for 25 epochs. After 25 epochs, there is not much change in loss means the loss function is converged. The best results using ANN are obtained for a buffer size of 50 meters. we can see performance metrics from Table 4.1.

After that, we trained hybrid models using deep learning models such as LSTM and GRU for temporal data and ANN for spatial data. The LSTM model takes data in sequences. The lag factor for time series models is taken as 4. After forming data sequences, the sequential data is given as input to LSTM and GRU. LSTMs and GRUs have the property that they form vector embeddings of temporal data after identifying temporal patterns in the data. We extracted these embeddings from the hidden layer. These embeddings are provided as input to ANN along with the spatial data. Now, ANN has three hidden layers containing eight units, 16 units, and eight units, respectively. The final output from ANN is the spatiotemporal predictions. The best results are obtained for a buffer size equal to 100 meters for the LSTM-ANN model and a buffer size equivalent to 50 meters for the GRU model.

Out of all models, the best performance metrics are obtained for the GRU-ANN model for a buffer size equal to 50 meters. It has an R squared value equal to 0.747. But it can be observed that LSTM-ANN results are consistent for all buffer sizes. We can see that LSTM-ANN has R-squared values of 0.707, 0.737, 0.71, and 0.72 for buffers of sizes 50 meters, 100 meters, 150 meters, and 200 meters.

Fig. 4.13 shows a comparison plot for different models' predictions. In this plot, we have used the best prediction values for each model. In this plot time, the bin is taken as 75 so that the prediction results can be seen easily. Fig. 4.14 shows a comparison plot of different models for test data of 10 days.

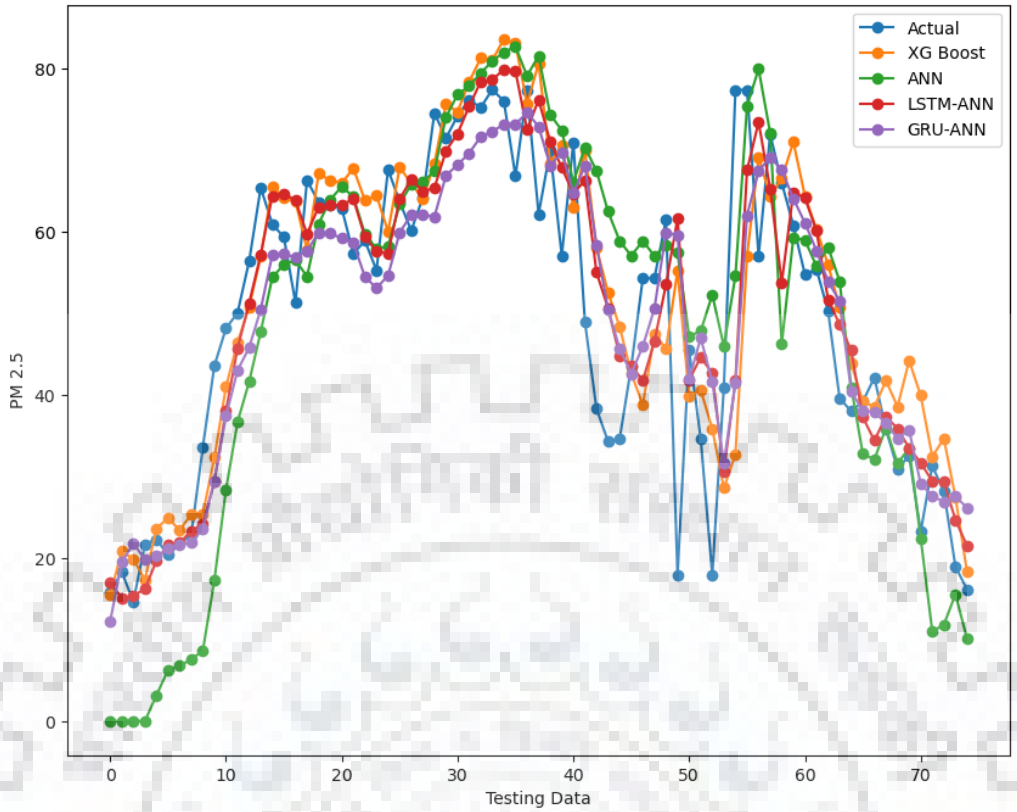

Figure 4.13: Comparison of predictions using different models on test data for time bin equals to 75

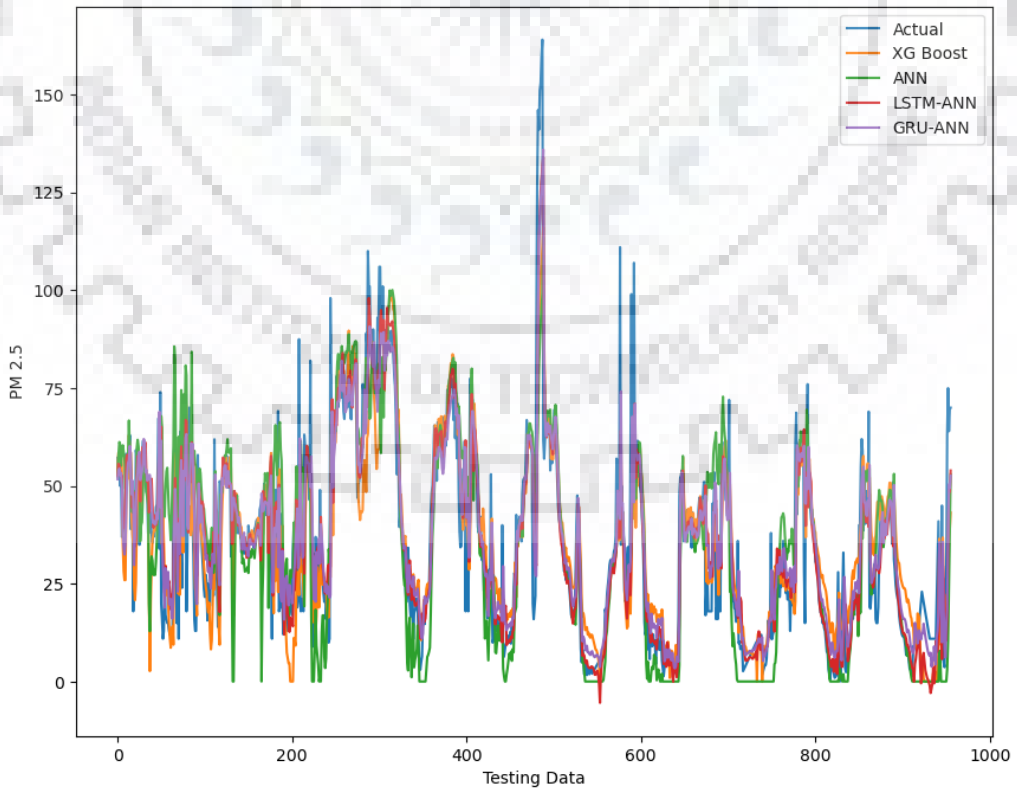

Figure 4.14: Comparison of predictions using different models on test data

Table 4.1: Performance Metrics for Different Models for different buffer size

Rectangular buffer of width 50 meters					
Model	**MSE**	**RMSE**	**MAE**	**MAPE**	**R squared**
XG Boost	231.84	15.226	10.73	0.742	0.636
ANN	199.02	14.1	9.37	0.5	0.687
LSTM-ANN	186.66	13.66	9.78	0.95	0.707
GRU-ANN	161.17	12.69	8.32	0.53	0.747
Rectangular buffer of width 100 meters					
Model	**MSE**	**RMSE**	**MAE**	**MAPE**	**R squared**
XG Boost	273.36	16.53	11.26	0.61	0.571
ANN	215.54	14.68	10.31	0.44	0.66
LSTM-ANN	167	12.92	8.38	0.47	0.737
GRU-ANN	192.02	13.85	9.11	0.43	0.698
Rectangular buffer of width 150 meters					
Model	**MSE**	**RMSE**	**MAE**	**MAPE**	**R squared**
XG Boost	259.38	16.1	12.13	1.18	0.592
ANN	212.76	14.58	10	0.41	0.666
LSTM-ANN	184.77	13.59	9.01	0.52	0.71
GRU-ANN	193.85	13.92	10.12	1.05	0.6957
Rectangular buffer of width 200 meters					
Model	**MSE**	**RMSE**	**MAE**	**MAPE**	**R squared**
XG Boost	235.12	15.33	11.16	1.1	0.631
ANN	211.81	14.55	10.22	0.9	0.667
LSTM-ANN	178.28	13.35	8.99	0.56	0.72
GRU-ANN	196.22	14	10.26	1.09	0.692

Chapter 5

Conclusion and Future work

5.1 Conclusion

In the given study, temporal variables such as $PM_{2.5}$ static, meteorological factors such as Wind Speed and Atmospheric Temperature, and length of roads divided according to congestion factor significantly impact the predictions of $PM_{2.5}$ mobile at locations without monitoring stations. To include the effect of road congestion, roads were divided into four categories according to the congestion factor.

Spatial factors also correlate with $PM_{2.5}$ mobile, but the impact of spatial features is very low for prediction. It can be seen from Fig. 4.5 that $PM_{2.5}$ mobile negatively correlates with green, residential, and water bodies areas.

From Table 4.1, we can analyze the performance of different models using the metric used in the table. Mean squared error(MSE) indicates the overall accuracy of the models, and it should be as low as possible. GRU-ANN has the lowest MSE when buffer widths are 50 meters. On the other hand, LSTM-ANN achieved the lowest MSE in the different 3 scenarios.

Mean absolute error (MAE) represents the average magnitude of the errors. GRU-ANN model showed the best performance when the buffer width was 50 meters. At the same time, LSTM-ANN achieved the lowest MAE in the other three scenarios.

The Mean Absolute Percentage Error (MAPE) measures the average percentage difference between the actual and predicted values. Lower values of MAPE are desired. ANN has the lowest MAPE when buffer widths are 50 and 150 meters. GRU-ANN has the lowest MAPE for a buffer width of 100 meters, and LSTM-ANN has the lowest MAPE for a buffer width of 200 meters.

GRU-ANN achieved the highest R squared values when the buffer width was 50 meters, and LSTM-ANN achieved the highest R squared values in all other scenarios.

It is observed that LSTM-ANN outperforms every other model for buffer widths of 100 meters, 150 meters, and 200 meters. Also, LSTM-ANN has consistent results for all four scenarios. Hence, LSTM-ANN is the most significant model of all the models used in the research.

5.2 Future work

In the present study, we have used training and validation data for six months and test data for ten days. Our models have yet to see all the temporal variations of an entire year. So, this model can be further trained for one-year data so that the model can see all the temporal variations of a year. This can result in better predictions of $PM_{2.5}$ mobile at locations without monitoring stations. Investigating the temporal dynamics and incorporating more granular temporal data in our model to capture short-term dependencies would be valuable. Additionally, since road congestion has been a crucial factor, further exploration can be done for better categorization of roads and to incorporate real-time traffic data to capture effects due to congestion better. Lastly, the potential of different models can be explored, and combining the strengths of other models could lead to improved predictive performance. By working in these areas in future research, we can improve $PM_{2.5}$ mobile predictions at locations without monitoring stations.

References

Chung, K. F., J. Zhang, and N. Zhong (2011). "Outdoor air pollution and respiratory health in Asia". In: *Respirology* 16.7, pp. 1023–1026 (cit. on p. 3).

Data, O. W. I. (2019). https://ourworldindata.org/outdoor-air-pollution (cit. on pp. iii, 2).

Diem, J. E. and A. C. Comrie (2002). "Predictive mapping of air pollution involving sparse spatial observations". In: *Environmental pollution* 119.1, pp. 99–117 (cit. on p. 7).

Eze, I. C., E. Schaffner, E. Fischer, T. Schikowski, M. Adam, M. Imboden, M. Tsai, D. Carballo, A. von Eckardstein, N. Künzli, et al. (2014). "Long-term air pollution exposure and diabetes in a population-based Swiss cohort". In: *Environment international* 70, pp. 95–105 (cit. on p. 2).

Gryech, I., Y. Ben-Aboud, M. Ghogho, and A. Kobbane (2021). "On spatial prediction of urban air pollution". In: *2021 17th International Conference on Intelligent Environments (IE)*. IEEE, pp. 1–6 (cit. on pp. v, 12).

Gryech, I., M. Ghogho, H. Elhammouti, N. Sbihi, and A. Kobbane (2020). "Machine learning for air quality prediction using meteorological and traffic related features". In: *Journal of Ambient Intelligence and Smart Environments* 12.5, pp. 379–391 (cit. on p. 12).

Guttikunda, S. K., K. Nishadh, and P. Jawahar (2019). "Air pollution knowledge assessments (APnA) for 20 Indian cities". In: *Urban Climate* 27, pp. 124–141 (cit. on p. 3).

Hoek, G., R. Beelen, K. De Hoogh, D. Vienneau, J. Gulliver, P. Fischer, and D. Briggs (2008). "A review of land-use regression models to assess spatial variation of outdoor air pollution". In: *Atmospheric environment* 42.33, pp. 7561–7578 (cit. on p. 9).

Hsieh, H.-P., S.-D. Lin, and Y. Zheng (2015). "Inferring air quality for station location recommendation based on urban big data". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 437–446 (cit. on p. 4).

Larkin, A., A. van Donkelaar, J. A. Geddes, R. V. Martin, and P. Hystad (2016). "Relationships between changes in urban characteristics and air quality in East Asia from

2000 to 2010". In: *Environmental science & technology* 50.17, pp. 9142–9149 (cit. on p. 3).

Larkin, A., J. A. Geddes, R. V. Martin, Q. Xiao, Y. Liu, J. D. Marshall, M. Brauer, and P. Hystad (2017). "Global land use regression model for nitrogen dioxide air pollution". In: *Environmental science & technology* 51.12, pp. 6957–6964 (cit. on p. 10).

Manisalidis, I., E. Stavropoulou, A. Stavropoulos, and E. Bezirtzoglou (2020). "Environmental and health impacts of air pollution: a review". In: *Frontiers in public health*, p. 14 (cit. on pp. 1, 2).

Manojkumar, N, M Monishraj, and B Srimuruganandam (2021). "Commuter exposure concentrations and inhalation doses in traffic and residential routes of Vellore city, India". In: *Atmospheric Pollution Research* 12.1, pp. 219–230 (cit. on p. 6).

Samal, K, K. Babu, and S. Das (2021). "Spatio-temporal prediction of air quality using distance based interpolation and deep learning techniques". In: *EAI Endorsed Transactions on Smart Cities* 5.14 (cit. on p. 13).

Soh, P.-W., J.-W. Chang, and J.-W. Huang (2018). "Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations". In: *Ieee Access* 6, pp. 38186–38199 (cit. on p. 8).

StackOverflow (2018). https://stackoverflow.com/questions/50488427/what-is-the-architecture-behind-the-keras-lstm-cell (cit. on pp. iii, 33).

Venn, A., S. Lewis, M. Cooper, R. Hubbard, I. Hill, R. Boddy, M. Bell, and J. Britton (2000). "Local road traffic activity and the prevalence, severity, and persistence of wheeze in school children: combined cross sectional and longitudinal study". In: *Occupational and environmental medicine* 57.3, pp. 152–158 (cit. on p. 7).

Wang, S., C. Fang, L. Sun, Y. Su, X. Chen, C. Zhou, K. Feng, and K. Hubacek (2019). "Decarbonizing China's urban agglomerations". In: *Annals of the American Association of Geographers* 109.1, pp. 266–285 (cit. on p. 3).

Wang, S., S. Gao, S. Li, and K. Feng (2020). "Strategizing the relation between urbanization and air pollution: Empirical evidence from global countries". In: *Journal of Cleaner Production* 243, p. 118615 (cit. on p. 3).

Wen, C., S. Liu, X. Yao, L. Peng, X. Li, Y. Hu, and T. Chi (2019). "A novel spatiotemporal convolutional long short-term neural network for air pollution prediction". In: *Science of the total environment* 654, pp. 1091–1099 (cit. on p. 11).

WHO (2008-09). https://www.who.int/news/item/02-05-2018-9-out-of-10-people-worldwide-breathe-polluted-air-but-more-countries-are-taking-action (cit. on p. 2).

Wikipedia (2011). https://en.wikipedia.org/wiki/Haversine_formula (cit. on p. 23).

Yu, R., Y. Yang, L. Yang, G. Han, and O. A. Move (2016). "RAQ–A random forest approach for predicting air quality in urban sensing systems". In: *Sensors* 16.1, p. 86 (cit. on pp. 8, 9).