# APPLICATION OF MACHINE LEARNING MODELS FOR ESTIMATION OF SPATIAL DISTRIBUTION OF PARTICULATE MATTER POLLUTANT IN AIR

**ALEX MWOLOLO KIMUYA**

**A Thesis Submitted in Partial Fulfilment for Conferment of the Degree of Master of Science in Physics of Meru University of Science and Technology**

**2025**

## DECLARATION

This thesis is my original work and has not been presented for award in any other institution.

SC409/201099/20

Signature_____ Date _____

**Alex Mwololo Kimuya**

## DECLARATION BY SUPERVISORS

This thesis has been submitted with our approval as the university supervisors.

Signature_____ Date _____

**Dr. Dickson Mwenda Kinyua, Ph.D**

Kirinyaga University, Kenya.

Signature_____ Date _____

**Dr. Daniel Maitethia Memeu, Ph.D**

Meru University of Science and Technology, Kenya.

## ACKNOWLEDGEMENT

I extend my deepest gratitude to my supervisors, Dr. Daniel Maitethia Memeu and Dr. Dickson Mwenda Kinyua, for their unwavering guidance and support throughout my MSc journey. Their expertise and encouragement were instrumental in shaping this research and fostering my academic growth. I am profoundly thankful to Madam Phylis Makena Mwenda for her invaluable assistance in preparing for my thesis defense. Her dedication and support were essential to its successful completion. I would also like to express my sincere appreciation to Mr. Bendibbie Mulwa for his insightful contributions and stimulating discussions. His perspectives enriched my research significantly. The camaraderie and support of my colleagues and friends have made this experience truly rewarding. I am especially grateful to those who shared this academic path with me. My family's unconditional love and support have been my constant source of strength. I am particularly indebted to my parents and siblings for their unwavering belief in me. I would like to acknowledge the contributions of the faculty and staff at Meru University of Science and Technology (MUST) to my academic journey. Their dedication to creating a conducive learning environment is commendable. Finally, I express my sincere thanks to everyone who, directly or indirectly, contributed to this thesis. Your support is deeply appreciated.

Thank you.

## ABSTRACT

Air pollution, particularly particulate matter, poses significant health and environmental risks. Conventional particulate matter measurement systems provide only single-point instantaneous readings, limiting their ability to generate spatial distribution maps. This study applies machine learning techniques to estimate particulate matter distribution using data from a limited number of sensor nodes. The sensors collected data on PM1.0, PM2.5, and PM10 concentrations, alongside weather parameters (wind speed, temperature, humidity) and spatial information (longitude, latitude). Various machine learning models, including Artificial Neural Networks, Support Vector Regression, Long Short-Term Memory, and Random Forest, were evaluated for particulate matter distribution prediction. The analysis revealed that Artificial Neural Networks consistently outperformed other models across various feature configurations for predicting particulate matter concentrations. When considering only geometric features (Euclidean_D and Orientation), the Artificial Neural Networks model achieved training $R^2$ scores ranging from 0.9746 to 0.9790 (with features: wind speed, temperature, geometric features) for PM1.0 prediction. Similar trends were observed for PM2.5 and PM10, with Artificial Neural Networks training $R^2$ scores between 0.9766 (geometric features only) and 0.9805 (humidity, geometric features) for PM2.5, and 0.9792 (geometric features only) to 0.9798 (wind speed, temperature, geometric features) for PM10. Validation $R^2$ scores remained impressive for Artificial Neural Networks, ranging from 0.9711 to 0.9756 (PM1.0), 0.9692 to 0.9789 (PM2.5), and 0.9775 to 0.9793 (PM10) under the respective particulate matter scales feature combinations. For generalizability on unseen data, Artificial Neural Networks also achieved the highest average prediction accuracy, exceeding 75% consistently across all feature combinations for PM1.0, PM2.5, and PM10 concentrations within a 160-meter radius from a central sensor. Further, the findings revealed that incorporating the weather parameters significantly improved model performance, reducing both RMSE and MAE. Specifically, PM1.0 RMSE decreased from 1.8719 $\mu g/m^3$ to 1.7201 $\mu g/m^3$, and MAE from 0.8125 $\mu g/m^3$ to 0.7952 $\mu g/m^3$. For PM2.5, RMSE reduced from 2.5260 $\mu g/m^3$ to 2.2139 $\mu g/m^3$, and MAE from 1.1488 $\mu g/m^3$ to 1.0012 $\mu g/m^3$. Lastly, PM10 RMSE decreased from 2.231 $\mu g/m^3$ to 2.143 $\mu g/m^3$, and MAE from 1.1120 $\mu g/m^3$ to 1.0037 $\mu g/m^3$. This research demonstrates a machine learning-based approach to overcome limitations of single-point particulate matter sensors and predict particulate matter distribution across a region. The research results highlight the effectiveness of Artificial Neural Networks, achieving high accuracy and emphasizing the value of including weather data in model training for improved particulate matter distribution prediction. The proposed approach offers a practical solution for real-world air quality monitoring by enabling data-driven decision-making in pollution management, optimizing sensor placement strategies, and facilitating more accurate identification of pollution hotspots for targeted interventions. The study findings suggest a new approach to designing particulate matter sensors, overcoming limitations of single-point measurement, and emphasize the importance of including weather parameters in machine learning model training for spatial distribution prediction.

## ACRONYMS AND ABREVIATIONS

| | |
|---|---|
| ANN | Artificial Neural Network |
| AMQ | Air Quality Monitoring |
| CNNs | Convolutional neural networks |
| GBMs | Gradient boosting machines |
| GP | Gaussian Processes |
| IDW | Inverse Distance Weighting |
| IoT | Internet of Things |
| LSTM | Long Short-Term Memory |
| MAE | Mean Absolute Error |
| ML | Machine Learning |
| PM | Particulate Matter |
| PM1.0 | Particulate Matter ≤ 1.0 μm |
| PM2.5 | Particulate Matter ≤ 2.5 μm |
| PM10 | Particulate Matter ≤ 10 μm |
| PoI | Point of Interest |
| $R^2$ | Coefficient of Determination |
| RBF | Radial basis function |
| RF | RandomForest |
| RMSE | Root Mean Squared Error |
| RNNs | Recurrent neural networks |
| SVR | Support Vector Regression |
| VOCs | Volatile organic compounds |
| WiFi | Wireless Fidelity |
| WHO | World Health Organization |

## TABLE OF CONTENTS

**DECLARATION** ..... **II**
**ACKNOWLEDGEMENT** ..... **III**
**ABSTRACT** ..... **IV**
**ACRONYMS AND ABREVIATIONS** ..... **V**
**TABLE OF CONTENTS** ..... **VI**
**LIST OF FIGURES** ..... **VIII**
**LIST OF TABLES** ..... **IX**
**LIST OF APPENDICES** ..... **X**
**CHAPTER ONE: INTRODUCTION** ..... **1**
1.1 Research Background ..... 1
1.2 Problem Statement ..... 5
1.3 Purpose of the Study ..... 7
1.4 Research Objectives ..... 7
1.4.1 Main objective ..... 7
1.4.2 Specific objectives ..... 7
1.5 Justification of the Study ..... 7
1.6 Significance of the Study ..... 9
1.7 Scope of the Study ..... 9
1.8 Limitations and Delimitations of the Study ..... 10
1.8.1 Limitations of the study ..... 10
1.8.2 Delimitations of the research ..... 10
**CHAPTER TWO: LITERATURE REVIEW** ..... **12**
2.1 Introduction ..... 12
2.2 Particulate Matter and its Health Implications ..... 12
2.3 Environmental Parameters and Air Pollution ..... 14
2.4 Spatial Distribution Inference Techniques ..... 16
2.5 Sensor Technology and Sensing Principles ..... 19
2.6 Machine Learning Models in Air Quality Prediction ..... 21
2.7 Integrative Approaches for Air Quality Monitoring ..... 23
2.8 Summary and Research Gaps ..... 25
**CHAPTER THREE: METHODOLOGY** ..... **27**
3.1 Introduction ..... 27
3.2 The Study Site ..... 27
3.3 Sensor System Design and PM Concentration Data Collection Process ..... 28
3.3.1 PMS5003 sensor principles and operation ..... 29
3.3.2 Design of the cloud-based server ..... 32
3.3.3 Data collection process ..... 32
3.3.4 Dataset preparation and preprocessing ..... 34
3.4 Training and Evaluation Strategy ..... 38
3.4.1 Signal energy analysis ..... 42
**CHAPTER FOUR: RESULTS AND DISCUSSION** ..... **46**
4.1 Introduction ..... 46
4.2 Pre-Deployment Analysis ..... 46
4.2.1 Machine leaning models training and validation for PM1.0 ..... 47
4.2.2 Machine leaning models training and validation for PM2.5 ..... 49
4.2.3 Machine leaning models training and validation for PM10 ..... 52
4.2.4 Average percentage prediction accuracy analysis ..... 54
4.2.5 Feature importance ..... 64

4.2.6 Validating the ANNS model as the best performer ..... 72
4.3 Assessment of Models' Prediction Accuracies with Distance ..... 74
4.3.1 Post-deployment results ..... 75
4.3.2 Post-deployment results discussion ..... 82
**CHAPTER FIVE: CONCLUSION, RECOMMENDATIONS AND PUBLICATION ..... 86**
5.1 Conclusion ..... 86
5.2 Recommendations ..... 88
5.3 Publication ..... 91
**REFERENCES ..... 92**
**APPENDICES ..... 113**

## LIST OF FIGURES

Figure 3.1: Sketch map of the Study Site-Meru University of Science and Technology. ..... 28
Figure 3.2: Flow of laser scattering in PMS5003 Sensor. ..... 30
Figure 3.3: Illustration of custom-designed IoT enabled sensor system for measuring PM concentrations. ..... 32
Figure 3.4: Illustration of the IoT enabled sensor system user dashboards. ..... 34
Figure 3.5: Application of Median Filtering to Remove Spurious Values in the Raw Data. ..... 38
Figure 4.1: Prediction Accuracy of ANN Model for PM1.0 Concentrations across a160-Meter Radius. ..... 76
Figure 4.2: Smoothened PM1.0 ANN Prediction Accuracy with Change in Distance. 77
Figure 4.3: Prediction Accuracy of ANN Model for PM2.5 Concentrations across a 160-Meter Radius from Central Sensor Node. ..... 78
Figure 4.4: Smoothened PM2.5 ANN Prediction Accuracy with Change in Distance. 79
Figure 4.5: Prediction Accuracy of ANN Model for PM10 Concentrations across a 160-Meter Radius from Central Sensor Node. ..... 81
Figure 4.6: Smoothened PM10 ANN Prediction Accuracy with Change in Distance. . 82

## LIST OF TABLES

Table 4.1: Evaluation of Machine Learning Models for PM1.0 Prediction. ..... 47
Table 4.2: Impact of Feature Selection on PM2.5 Prediction Model Performance. ..... 51
Table 4.3: Impact of Feature Combinations on PM10 Prediction Model Performance. 54
Table 4.4: ANN Model Average Prediction Accuracy for PM1.0, PM2.5, and PM10. 56
Table 4.5: SVR Model Average Prediction Accuracy for PM1.0, PM2.5, and PM10. 58
Table 4.6: LSTM Model Average Prediction Accuracy for PM1.0, PM2.5, and PM10. 60
Table 4.7: RandomForest Model Average Prediction Accuracy for PM1.0, PM2.5, and PM10. ..... 63
Table 4.8: Feature Importance Ranking for PM1.0 Prediction Models. ..... 65
Table 4.9: Feature Importance Ranking for PM2.5 Prediction Models. ..... 68
Table 4.10: Feature Importance Ranking for PM10 Prediction Models. ..... 70

## LIST OF APPENDICES

Appendix I: Implementation of the ANNs PM Model ..... 114
Appendix II: Implementation of SVR PM Model ..... 118
Appendix III: Implementation of LSTM PM Model ..... 121
Appendix IV: Implementation of Random Forest Regression PM Model ..... 125
Appendix V: Publication ..... 128
Appendix VI: Plagiarism Report ..... 129

# CHAPTER ONE: INTRODUCTION

## 1.1 Research Background

Air pollution, characterized by the presence of particulate matter (PM) in the atmosphere, poses a significant global threat to health and the environment (Manisalidis et al., 2020a; Schraufnagel, 2020; Mukherjee and Agrawal, 2018). PM encompasses microscopic solid and liquid particles suspended in the air, with detrimental effects on human health, ecosystems, and climate (Thangavel et al., 2022; Ukaogo et al., 2020).

These particles originate from diverse sources such as industrial emissions, vehicle exhaust, biomass burning, and dust (Ukaogo et al., 2020). PM is classified into three main scales including; PM10, PM2.5, and PM1.0. PM10 includes particles with a diameter of 10 micrometers or smaller, PM2.5 includes even finer particles with a diameter of 2.5 micrometers or smaller, and PM1.0 consists of the smallest and most respirable particles, with a diameter of 1.0 micrometer or smaller (Abulude et al., 2022; Alfano et al., 2020a).

Their small size allows them to penetrate deep into the respiratory system, causing serious health risks upon inhalation such as respiratory and cardiovascular diseases (Manisalidis et al., 2020a; Crinnion, 2017; Kelly and Fussell, 2015). Given these health implications, accurate measurement of PM concentrations with high degree of spatial-temporal resolution should be key consideration in deployment of PM monitoring stations.

Conventional particulate matter (PM) measurement systems typically rely on single spatial point measurements, neglecting the mapping of PM spatial distribution. Obtaining geo-spatial PM data often requires deploying multiple monitoring stations, which can be expensive and logistically challenging when high spatial resolution and high accuracy estimates are required. Interpolation methods have been utilized to estimate PM

distribution within regions of interest based on sensor measurements (Choi and Chong, 2022; Li et al., 2014, 2015). However, existing methodologies often overlook the complex relationships between environmental factors (wind speed, temperature, and humidity) and the spatiotemporal distribution of PM concentrations (Giordano et al., 2021; Li et al., 2022; Narayana et al., 2022; Sabir et al., 2024; Shiferaw et al., 2023). Integrating these environmental parameters into PM distribution models is important, yet challenging due to their complex interplay. Developing practical analytical models that incorporate these environmental factors remains a formidable task.

The adoption of Internet-of-Things (IoT) systems for particulate matter (PM) monitoring has revolutionized air quality assessment by enabling real-time data collection and remote accessibility. Low-cost PM sensors, such as optical particle counters and laser-based scattering sensors, have significantly enhanced the feasibility of deploying dense sensor networks for improved spatial and temporal monitoring of air pollution (Munir et al., 2019; Narayana et al., 2022). These sensors provide cost-effective alternatives to conventional gravimetric and beta attenuation monitoring techniques, which are often expensive and geographically sparse.

However, a common limitation of low-cost PM sensors is their susceptibility to environmental interferences, including humidity and temperature variations, which can affect measurement accuracy (Hagan et al., 2018). This limitation necessitates the virtualization of PM monitoring techniques through data-driven approaches, such as machine learning (ML) models, to enhance accuracy and reliability. ML technologies facilitate sensor calibration, error correction, and spatiotemporal PM estimation by integrating diverse environmental variables (Guo et al., 2022a; Liang et al., 2020). By leveraging ML-driven approaches, IoT-based PM monitoring systems can improve air

pollution mapping, ensuring more precise assessments of PM exposure levels across different regions.

Further, while previous studies (Chen et al., 2021; Guo et al., 2022b; Liang et al., 2020; Mehmood et al., 2022; Méndez et al., 2023; Rybarczyk and Zalakeviciute, 2018; Sharma et al., 2021; Zaman et al., 2021) successfully utilized machine learning models to capture air quality patterns and predict pollutant concentrations, they often faced limitations due to the data used. These studies primarily relied on data collected from individual monitoring stations or small geographic regions. This localized approach presents two key challenges. First, the generalizability of predictions is limited. Because the models are trained on data from specific locations, their accuracy can suffer when applied to broader areas. Extrapolating from a single point to a larger region fails to capture the important spatial variations in air quality that exist across landscapes. This can lead to the creation of misleading pollution maps that do not accurately reflect the true distribution of pollutants.

Second, there is an incomplete understanding of how different machine learning models perform under various factors. Current research has not explored how model effectiveness changes with the inclusion of diverse data points. This includes individual or combined meteorological parameters (wind speed, humidity, temperature) and spatial factors like topography or land cover. Hence, a more comprehensive understanding is crucial for identifying the best machine learning models for specific air quality prediction scenarios. Furthermore, existing research has limitations.

While some studies utilize feature engineering to capture complex pollution distributions based on localized measurements (Sabir et al., 2024; Shiferaw et al., 2023; Li et al., 2022), none have focused on feature engineering based on the separation distances between PM monitoring stations within a network. This presents a gap in knowledge, as

the distance between sensors likely influences the model's ability to predict PM distribution across an area.

This research addresses these limitations in air quality prediction by incorporating a wider range of environmental data into the models. This approach leads to more robust and generalizable models that account for the complex interplay between pollutants, weather conditions, and spatial variations.

Recent advancements in machine learning (ML) have shown significant promise in addressing the challenges of air quality monitoring and spatial distribution estimation of PM, (Guo et al., 2022a; Liang et al., 2020; Mehmood et al., 2022; Méndez et al., 2023; Rybarczyk and Zalakeviciute, 2018; Sharma et al., 2021; G. Zhang et al., 2018a). This study investigates the potential of four prominent ML models for predicting PM concentrations: Long Short-Term Memory (LSTM) networks, Artificial Neural Networks (ANNs), Support Vector Regression (SVR), and Random Forest (RF) models.

LSTMs are particularly well-suited for analyzing the time-series nature of PM concentration data, allowing them to learn from sequential patterns and improve prediction accuracy (Chen et al., 2021; Xu et al., 2022; Zhang et al., 2018b). ANNs are known for their adaptability and ability to capture complex relationships within data, making them a valuable tool for modeling air quality, as demonstrated in studies focused on PM prediction (Goulier et al., 2020). SVR models have proven effective in capturing non-linear relationships between variables, a crucial aspect of air quality modeling due to the complex interplay between pollutants, weather conditions, and spatial variations (Kshirsagar and Khare, 2023; Sánchez et al., 2011).

Finally, Random Forest models are recognized for their robustness to noise in data and their ability to handle high-dimensional datasets, which are common characteristics of air quality monitoring data (Saragih and Mazdadi, 2023; Tella et al., 2021; Zamani

Joharestani et al., 2019). By evaluating the performance of these four distinct ML models, this study aims to identify the most effective approach for incorporating a wider range of environmental data into air quality prediction models.

The study developed an integrated machine learning approach to predict the spatial distribution of PM within a specified radius. This approach utilized data from limited local sensor measurements, including PM concentrations of various sizes (PM1.0, PM2.5, and PM10) alongside important environmental parameters like wind speed, temperature, and humidity. Additionally, the sensor network incorporated data on each sensor's separation distance (Euclidean_D) and Orientation relative to a central sensor node.

The analysis of the complex relationships between PM concentrations, weather patterns, and sensor location provides valuable insights not only for improving air quality models but also for real-world applications. Environmental watchdogs can leverage this approach to remotely locate PM sources. Through understanding how PM concentrations vary with weather patterns and sensor location, these groups can identify areas with potentially high emissions. This information can then be used to quantify emissions and set more effective permissible emission limits, ultimately leading to better air quality management strategies.

**1.2 Problem Statement**

Air pollution, specifically particulate matter (PM), is a major threat to human health and the environment. PM is a complex mixture of tiny particles and droplets suspended in the air, and its size, composition, and concentration can vary greatly. Understanding and monitoring PM levels are essential for evaluating air quality, reducing health risks, and implementing effective pollution control. However, current PM monitoring systems have significant limitations. They typically rely on single-point measurements taken at specific locations. This approach provides limited spatial coverage and fails to capture

how PM concentrations change across an area. This makes it difficult to identify pollution sources and develop targeted control strategies.

Additionally, existing monitoring networks are often sparse, leaving large data gaps, especially in wider regions. This hinders accurate air quality assessment and makes it challenging to map PM distribution without taking independent measurements in many locations. Deploying dense networks of low-cost PM sensors can be expensive, logistically complex, or impractical due to safety concerns or limited accessibility. While machine learning (ML) models have shown promise in predicting PM concentrations, past research has focused on their instantaneous accuracy without exploring their ability to infer spatial information based on proximity to a reference point. This limits the ability to fully utilize ML models for accurate spatial analysis.

Furthermore, current ML model research has limitations in exploring how the distance between monitoring stations affects model accuracy, how incorporating distance-based features and weather data during training can improve performance, and the relationship between feature importance in ML models and the separation distance between monitoring stations. Therefore, the primary objective of this study was to develop a PM sensing system that goes beyond the limitations of single-point monitoring, addresses data gaps, explores the spatial capabilities of ML models, and leverages reference sensor nodes for accurate spatial inference of PM distribution. This research sought to bridge these gaps and contribute valuable insights to air quality assessment and management, ultimately leading to more effective monitoring and control of particulate matter pollution.

### 1.3 Purpose of the Study

The purpose of this research was to apply machine learning models, using limited sensor data and weather parameters, to estimate the spatial distribution of particulate matter air pollution.

### 1.4 Research Objectives

#### 1.4.1 Main objective

The primary objective of this research is to apply machine learning models for the estimation of particulate matter pollutants in air and infer their spatial distribution within a specified radius.

#### 1.4.2 Specific objectives

i. To design an Internet of Things (IoT)-enabled sensor system for accurate measurements of particulate matter concentrations, spatial data (Latitude and Longitude), and meteorological parameters (Wind speed, ambient Temperature, and Humidity) at multiple spatial points.

ii. To evaluate machine learning models, including Long Short-Term Memory (LSTM), Artificial Neural Networks (ANN), Support Vector Regression (SVR), and RandomForest, for predicting the spatial distribution of PM pollutants.

iii. To assess the impact of integrating meteorological parameters (Wind speed, Temperature, and Humidity) as input features in training machine learning models for predicting the spatial distribution of particulate matter concentration within a specified geographic area, using limited local measurements.

### 1.5 Justification of the Study

One major limitation addressed is the reliance on single-point monitoring, a widely used but inadequate approach (Alfano et al., 2020a; Ardon-Dryer et al., 2020; Bucek et al., 2021; deSouza et al., 2020; Wallace & Hopke, 2022). This method provides a limited

view of PM distribution, hindering the identification of pollution sources and the development of targeted control strategies. This study proposes a novel approach that goes beyond single-point monitoring, offering a more comprehensive understanding of PM distribution.

Furthermore, the study addresses the issue of sparse monitoring networks, a common problem leading to data gaps and hindering accurate air quality assessments across large areas. The research explores an IoT-enabled sensor system designed for dense data collection at multiple spatial points. This approach promises to fill data gaps and provide a deeper understanding of PM distribution patterns, ultimately improving air quality management decisions.

This research also delves into a previously unexplored area: the spatial capabilities of machine learning models for PM prediction. By systematically examining this aspect, the study offers valuable insights into how ML models can be effectively used for spatial inference. This knowledge can revolutionize air quality modeling and forecasting, enabling more accurate and localized assessments.

The study emphasizes the importance of incorporating weather data, such as wind speed, temperature, and humidity, into ML model training. This integration leads to more robust and accurate predictions by considering the influence of weather on pollutant dispersion. This knowledge offers valuable insights to researchers seeking to understand the factors influencing PM distribution.

The study's findings hold significant value for various stakeholders. Environmental watchdogs can leverage the identified pollution hotspots to target awareness campaigns, fostering community engagement in advocating for cleaner air. Policymakers can utilize the data to identify pollution hotspots for resource allocation and targeted Air Quality Management strategies. This data-driven approach fosters effective air quality policies

based on objective evidence and strengthens efforts to push for stricter regulations. Furthermore, the study demonstrates the potential for cost-effective solutions using limited sensor nodes and ML, a valuable insight for policymakers seeking efficient monitoring networks. Researchers benefit from established performance benchmarks that guide future research and continuous advancements in PM prediction.

### 1.6 Significance of the Study

This study addresses critical gaps in air quality monitoring by developing a machine learning framework to estimate particulate matter (PM) spatial distribution using a limited number of sensor nodes, providing a cost-effective alternative to dense sensor networks. The integration of meteorological parameters (wind speed, temperature, humidity) and geometric features (Euclidean distance, orientation) enhances the understanding of how environmental factors influence PM dispersion. The demonstrated superiority of Artificial Neural Networks (ANNs) over other models offers valuable insights for optimizing air quality predictions, particularly in resource-constrained regions. The findings support policymakers in identifying pollution hotspots, optimizing sensor placement, and designing targeted mitigation strategies. Additionally, the study's IoT-enabled sensor system and open-source Python implementation on Google Colab improve reproducibility, offering a scalable template for global air quality management and promoting advancements in smart environmental monitoring technologies.

### 1.7 Scope of the Study

This research focuses on predicting PM1.0, PM2.5, and PM10 concentrations within a 160-meter radius of a central sensor node at Meru University of Science and Technology, Kenya. It evaluates four machine learning models (ANNs, SVR, LSTM, Random Forest) using time-series data from IoT sensors, including PM measurements, GPS coordinates, and meteorological parameters. The scope encompasses preprocessing techniques

(median filtering, categorical encoding), feature engineering (Euclidean distance calculations, orientation encoding), and model validation via $R^2$, RMSE, and MAE metrics. While the study emphasizes spatial distribution, it does not address long-term temporal trends or chemical composition analysis of PM.

### 1.8 Limitations and Delimitations of the Study

#### 1.8.1 Limitations of the study

i. **Sensor Accuracy.** Reliance on low-cost PMS5003 sensors may introduce measurement biases under high humidity or extreme temperatures, affecting model inputs.

ii. **Geographic Specificity.** Data collection was limited to a single university campus, potentially reducing generalizability to regions with divergent topography or emission sources.

iii. **Temporal Constraints.** The three-week data collection period may not capture seasonal variations in PM concentrations or weather patterns.

iv. **Model Generalizability.** While ANNs outperformed other models, their efficacy in hyper-dense urban or remote rural settings remains untested.

#### 1.8.2 Delimitations of the research

i. **Model Selection.** The study focused on ANNs, SVR, LSTM, and Random Forest, excluding emerging architectures like transformers or graph neural networks.

ii. **Feature Prioritization.** Weather parameters were restricted to wind speed, temperature, and humidity; other factors (e.g., precipitation, solar radiation) were excluded.

iii. **Spatial Resolution.** Analysis was confined to a 160-meter radius, avoiding broader-scale regional PM transport dynamics.

iv. **Data Preprocessing.** Median filtering was prioritized for noise reduction, while alternative methods (e.g., wavelet transforms) were not explored.

# CHAPTER TWO: LITERATURE REVIEW

## 2.1 Introduction

This chapter reviews existing literature on air quality monitoring and particulate matter (PM) concentration estimation. It explores the health effects of different PM types, environmental factors influencing air pollution (wind speed, direction, temperature, humidity), and spatial distribution techniques (kriging, inverse distance weighting, Gaussian processes) for PM concentration estimation. The chapter also discusses sensor technology, sensing principles and limitations, and the application of machine learning models for air quality prediction. It explores integrative approaches combining sensor technology with machine learning for enhanced monitoring and synthesizes key findings from the literature to identify research gaps, justifying this study.

## 2.2 Particulate Matter and its Health Implications

Particulate matter (PM) is a complex mixture of tiny solid particles and liquid droplets suspended in the air. These particles come in a surprising range of sizes, compositions, and origins. Scientists categorize PM based on size, with the most studied categories being PM10 (particles with a diameter of 10 micrometers or less), PM2.5 (particles with a diameter of 2.5 micrometers or less), and PM1.0 (particles with a diameter of 1.0 micrometers or less) (Manisalidis et al., 2020b; Stanek and Brown, 2019; Ukaogo et al., 2020). Notably, a human hair is roughly 70 micrometers in diameter (Yang, et al., 2017), so PM2.5 and finer particles are incredibly small.

The sources of PM are equally diverse, originating from both natural and human activities. Natural sources include dust storms, wildfires, and volcanic eruptions. Human activities like combustion processes in vehicles and power plants, industrial emissions, construction dust, and even cooking with solid fuels contribute significantly to air pollution (Manisalidis et al., 2020b; Stanek and Brown, 2019).

The size and composition of PM particles determine the health risks they pose. Fine and ultrafine particles, particularly PM2.5 and PM1.0, are especially concerning because of their ability to penetrate deep into the lungs. Once inhaled, these tiny particles can lodge in the alveoli, the air sacs where gas exchange occurs, and even enter the bloodstream (Faour et al., 2022; Kelly & Fussell, 2015; Manisalidis et al., 2020b; Sadrizadeh et al., 2022). This deep penetration has severe health consequences. Short-term exposure to high PM levels can trigger acute respiratory symptoms like coughing, wheezing, and shortness of breath. It can also worsen existing respiratory conditions like asthma and bronchitis (Schraufnagel, 2020; Thangavel et al., 2022; Xing et al., 2016). Perhaps most alarmingly, long-term exposure to PM has been linked to the development of chronic respiratory diseases, cardiovascular problems, and even an increased risk of premature death (Boldo et al., 2006; Larson et al., 2022).

Particulate matter (PM) is a complex air pollutant with a diverse composition. It includes black carbon, organic compounds, heavy metals, and sulfates, all of which vary depending on the source. Industrial facilities and vehicle exhaust are major contributors to PM2.5 and even finer PM1.0, while construction dust and natural sources primarily release PM10 (Humans, 2016). The specific components of PM play a crucial role in its toxicity and health effects. Studies have shown that heavy metals and organic compounds pose a greater health risk than other components (Duffney et al., 2023; Humans, 2016; Kodzius et al., 2018).

Understanding the health impacts of PM necessitates accurate measurement and monitoring of its concentration in the air. Traditional methods like gravimetric techniques and beta attenuation systems have limitations. These methods often rely on single-point measurements and fail to capture the dynamic variations in PM concentration across an area (O'Connor et al., 2014; Wang et al., 2016). Fortunately,

advancements in sensor technology, particularly the rise of Internet-of-Things (IoT) based PM detection systems, offer significant improvements. These systems provide real-time data collection and remote monitoring capabilities, allowing for a more comprehensive assessment of exposure levels and facilitating the implementation of effective pollution control measures (Hagan et al., 2018; Munir et al., 2019; Narayana et al., 2022).

### 2.3 Environmental Parameters and Air Pollution

Environmental factors play a critical role in shaping air pollution dynamics, influencing the behavior, dispersion, and chemical transformation of pollutants, particularly particulate matter (PM). Wind speed and direction, ambient temperature, and humidity are some of the most significant parameters.

Wind patterns are fundamental to understanding how air pollution moves. Wind speed determines the dispersion and transport of pollutants. High wind speeds can dilute and disperse pollutants over large areas, reducing their local concentration, but potentially impacting distant locations (Aldaweesh, 2019; Ventura et al., 2013). Conversely, low wind speeds allow pollutants to accumulate in specific areas, exacerbating air quality issues. Wind direction, when considered alongside speed, helps identify pollution sources and track pollutant trajectories, which is crucial for targeted mitigation strategies (Lin et al., 2023; Qiao et al., 2022).

Ambient temperature also significantly impacts air pollution. Higher temperatures can enhance the formation of secondary pollutants like ozone and smog through reactions involving nitrogen oxides (NOx) and volatile organic compounds (VOCs) (Ventura et al., 2013; Wallace and Hopke, 2022). Additionally, temperature affects atmospheric stability, which in turn influences the vertical mixing of pollutants. Stable atmospheric conditions trap pollutants near the ground, leading to higher concentrations and worsening air

quality. Conversely, unstable conditions promote vertical mixing, dispersing pollutants and potentially reducing their concentration at the surface (Ventura et al., 2013).

Further, air quality is not simply dictated by the amount of pollutants present, but also by how they interact with the surrounding environment. Humidity, for instance, plays a crucial role in influencing the physical and chemical properties of particulate matter (PM). High humidity levels can cause hygroscopic growth in PM, meaning particles absorb water vapor and increase in size (Cheng et al., 2015; Won et al., 2021). This growth not only affects how quickly particles settle (sedimentation rate) but also their ability to become cloud condensation nuclei (CCN), influencing local weather patterns that can further impact air quality. Additionally, the interplay between humidity and PM can alter the particles' ability to scatter light, thereby affecting visibility and the overall radiative balance of the atmosphere (Won et al., 2021; Zang et al., 2019).

The influence of weather extends beyond humidity. Wind speed and direction significantly impact all PM sizes (PM10, PM2.5, and PM1.0) by dispersing and diluting them. However, this effect is more pronounced for coarser particles like PM10 due to their larger mass and faster settling rates (Kim et al., 2015; Kwok, et al., 2017). Ambient temperature also plays a role, particularly for fine PM (PM2.5 and PM1.0). Higher temperatures generally lead to increased concentrations of these fine particles due to enhanced photochemical reactions that create them (Hidy, 2019; Huy, 2015; Rxam, 2019). Humidity's interaction with PM is more complex. While it significantly affects PM2.5 and PM1.0 by promoting hygroscopic growth and altering their composition, its impact on PM10 is less direct but still influences particle size and deposition rates (Won et al., 2021; Zhang et al., 2023).

## 2.4 Spatial Distribution Inference Techniques

Spatial distribution inference techniques are essential for estimating the distribution of particulate matter (PM) concentrations across geographical areas, aiding in effective air quality management and public health protection. Among the commonly utilized techniques, geostatistical methods such as kriging play a pivotal role. Kriging is a spatial interpolation method that considers the spatial correlation between sampled points to estimate values at unsampled locations (Kleijnen, 2017; Sajid et al., 2013). By modeling the spatial dependence structure, kriging provides estimates of PM concentrations with uncertainty quantification, making it valuable for generating high-resolution PM concentration maps (Kleijnen, 2017).

Inverse Distance Weighting (IDW) is another widely applied interpolation technique in environmental studies, including air quality research. IDW estimates values at unsampled locations based on the inverse of distances to neighboring sample points, assuming that closer points have more influence on estimation (Al-husban, 2021; Maleika, 2020). While IDW is straightforward and computationally efficient, it may oversmooth or undersmooth PM concentration estimates and does not explicitly account for the spatial correlation between samples.

Gaussian Processes (GP) offer a probabilistic framework for spatial interpolation that models the spatial variability of PM concentrations based on observed data points (Davies et al., 2022; Lutman et al., 2004). GPs are flexible in capturing complex spatial patterns and provide uncertainty estimates for predicted values, which is crucial for assessing the reliability of PM concentration maps (Lutman et al., 2004). However, Gaussian Processes can be computationally intensive, particularly for large datasets, and require careful parameterization to achieve optimal performance.

More recent advancements include machine learning approaches for spatial inference of PM concentrations. Machine learning models, such as neural networks, random forests, and support vector regression, have been increasingly applied to predict PM distribution patterns (Ebrahimi-Khusfi et al., 2021; Liang et al., 2020). These models leverage the relationships between PM concentrations and various spatial predictors (e.g., meteorological data, land use) to improve the accuracy of spatial predictions beyond traditional interpolation methods (Li et al., 2016; Munyati and Sinthumule, 2021; Zhang et al., 2018b). Machine learning techniques offer the advantage of handling nonlinear relationships and complex interactions among predictors, enhancing the spatial resolution and predictive performance of PM concentration maps (Chhajer et al., 2022; Kim et al., 2022; Wong et al., 2021).

Despite their advantages, spatial distribution inference techniques face challenges, particularly in regions with sparse monitoring networks or heterogeneous pollutant sources. The effectiveness of these methods relies on the density and distribution of monitoring stations, with sparse data potentially leading to biased estimates and uncertainties in PM concentration maps (P. Kumar et al., 2015; Rohajawati et al., 2024). Moreover, techniques like kriging and Gaussian Processes assume stationarity in spatial processes, which may not hold true in areas with diverse landscapes or complex emission sources, necessitating adaptive approaches for accurate PM mapping (Bronowicka-Mielniczuk et al., 2019; Munyati and Sinthumule, 2021).

Furthermore, distance-based feature engineering techniques, particularly those involving spatial coordinates and separation distances between monitoring stations, play a crucial role in enhancing the accuracy and interpretability of spatial distribution inference models for particulate matter (PM) concentrations. While existing studies often focus on incorporating geographical features such as land use or meteorological data into machine

learning models for PM prediction (Chai et al., 2022; Yu et al., 2023), the direct integration of spatial distances between monitoring stations has been relatively overlooked.

One effective approach involves converting spatial coordinates (latitude and longitude) of monitoring stations into radian degrees and calculating separation distances between pairs of stations. This method accounts for the actual geographic distance between monitoring points, which can significantly influence PM dispersion patterns due to local emission sources, terrain features, and atmospheric conditions (Agramanisti Azdy and Darnis, 2020; Siahaan, 2017). By including these distance-based features, spatial interpolation models can better capture localized variations in PM concentrations that may be missed by traditional interpolation techniques.

Research has demonstrated the efficacy of spatially informed models in assessing PM2.5 concentrations and analyzing pollution patterns near urban traffic sources. For instance, (Meng et al., 2021) applied a machine-learning-based Spatial Distribution Model (SDC) to estimate PM2.5 concentrations around residential buildings located in high-traffic urban areas. This model effectively incorporated proximity to the road as a key feature, allowing it to capture nuanced horizontal and vertical pollution gradients relative to the roadway. Through on-site measurements and Computational Fluid Dynamics (CFD) simulations, the study validated the SDC model, observing a substantial decrease in PM2.5 levels—from 73 $\mu g/m^3$ near the road to a consistent range of 40–45 $\mu g/m^3$ at further distances and heights. Despite these advances, the study did not incorporate raw GPS data or degree-to-meter conversions, relying instead on predetermined distances in meters. Incorporating such geographic coordinates, as well as weather parameters like temperature, humidity, and wind patterns, could further improve model precision, as these factors influence pollution dispersion. Expanding spatially informed ML models

with these data-driven enhancements could significantly strengthen pollution assessment and urban health planning.

Moreover, distance-based feature engineering complements other geographical attributes by providing additional context on spatial relationships and connectivity between monitoring stations. This approach is particularly beneficial in regions with sparse monitoring networks, where traditional interpolation methods may struggle to accurately represent PM distribution patterns. By quantifying the spatial separation between monitoring sites, distance-based features contribute to a more comprehensive understanding of local-scale PM dynamics and facilitate targeted interventions for air quality management.

### 2.5 Sensor Technology and Sensing Principles

Sensor technology for air quality monitoring has undergone significant evolution, driven by advancements in electronics, materials science, and environmental monitoring techniques. These developments have led to sensors that are more compact, sensitive, and capable of real-time data acquisition, essential for monitoring particulate matter (PM) pollution effectively (Bucek et al., 2021; Karagulian et al., 2019; Kumar and Gurjar, 2019; Molina Rueda et al., 2023). Modern sensors integrate various principles such as optical scattering, light absorption, and electrochemical sensing to detect PM concentrations across different size ranges, including coarse (PM10), fine (PM2.5), and ultrafine particles (PM1.0). These sensors not only provide continuous measurements but also offer high temporal and spatial resolution, crucial for understanding short-term fluctuations and spatial variability in PM levels (He et al., 2019; Rosenberg et al., 2012).

The PMS5003 sensor is a notable example of advanced sensor technology used for PM monitoring. Operating on light scattering principles, the PMS5003 employs a laser light source to illuminate particles passing through a detection chamber. The scattered light is

then detected and analyzed to determine the concentration of particles in different size categories (He et al., 2019). This sensor is preferred for its accuracy in measuring PM2.5 and PM10 concentrations in ambient air, offering real-time data that is essential for assessing air quality and supporting public health interventions.

Comparatively, traditional PM measurement techniques like gravimetric methods, beta attenuation monitors (BAMs), and DustTrak monitors have inherent limitations. Gravimetric techniques involve collecting particles on a filter medium and weighing them to estimate PM concentrations, but they can be time-consuming and prone to biases from humidity and particle bounce (O'Connor et al., 2014). BAMs are effective for PM2.5 measurements but lack sensitivity to ultrafine particles and may be affected by instrument drift over time (Raja et al., 2016). DustTrak monitors, based on light scattering, offer real-time measurements but are limited by their lower detection limits and inability to differentiate particle sizes effectively (Javed and Guo, 2021).

Despite their advancements, current sensor technologies face challenges that impact their reliability and applicability. These include calibration drift, cross-sensitivity to environmental factors such as humidity and temperature, and the need for frequent maintenance to ensure accurate readings (Karagulian et al., 2019). Furthermore, sensors deployed in outdoor environments must contend with weather-related factors like precipitation and extreme temperatures, which can affect their performance and longevity. Addressing these challenges requires ongoing research and development efforts to enhance sensor robustness, calibration procedures, and data validation techniques, ensuring the accuracy and reliability of PM measurements for air quality management and public health protection.

## 2.6 Machine Learning Models in Air Quality Prediction

Machine learning (ML) has emerged as a powerful tool in environmental monitoring, particularly for predicting air quality parameters such as particulate matter (PM) concentrations. ML models offer the capability to handle complex datasets and capture nonlinear relationships between environmental variables and pollutant levels. Various ML techniques have been applied in air quality prediction, including regression models, neural networks, decision trees, and ensemble methods (Liang et al., 2020; Mehmood et al., 2022; Rybarczyk & Zalakeviciute, 2018). These models leverage historical data from monitoring stations to learn patterns and make predictions, enabling real-time assessment of air quality and supporting decision-making processes.

Regression models, such as linear regression and support vector regression (SVR), are commonly used for predicting PM concentrations based on meteorological data, emission sources, and geographical factors (Rybarczyk & Zalakeviciute, 2018; Sánchez et al., 2011). Neural networks, including deep learning architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), excel in capturing complex spatial and temporal patterns in air quality data (Duan et al., 2023). Decision tree-based methods like random forests and gradient boosting machines (GBMs) are preferred for their ability to handle nonlinear relationships and feature interactions effectively (Jiang et al., 2021).

Studies implementing ML for PM concentration estimation have shown promising results in improving prediction accuracy compared to traditional statistical methods. For instance, (Xiao et al., 2020) applied a deep learning model to predict PM2.5 concentrations in Beijing, achieving higher accuracy by integrating satellite imagery and meteorological data. Similarly, (Xiao et al., 2018) utilized ensemble models to predict

PM2.5 levels in urban areas, demonstrating significant improvements in forecasting precision.

Despite these advancements, several limitations persist in current ML-based air quality prediction systems. One critical gap is the insufficient exploration of how the distance between monitoring stations impacts model accuracy. Distance-based features, such as separation distances and spatial relationships between sensors, are often overlooked in model training, potentially limiting the models' ability to generalize across different geographical settings. Incorporating distance-based features and weather data during model training could enhance performance by capturing local variations in pollutant dispersion and meteorological influences more accurately.

Moreover, the relationship between feature importance in ML models and separation radii between monitoring stations remains underexplored. Understanding which environmental factors and spatial parameters contribute most significantly to model predictions can guide the design of more robust air quality forecasting systems. For instance, (Wang et al., 2024) emphasized the need to integrate spatial context into feature engineering, showing that including distance-based features improved the interpretability and predictive power of their ML models.

Hence, while ML models have shown considerable promise in air quality prediction, addressing these research gaps is crucial for advancing the field. Future studies should focus on integrating distance-based features, exploring the impact of spatial relationships on model accuracy, and enhancing the interpretability of ML models in environmental monitoring applications. By leveraging these insights, researchers can develop more reliable and effective tools for managing air pollution and protecting public health in urban and industrial environments.

## 2.7 Integrative Approaches for Air Quality Monitoring

Integrative approaches for air quality monitoring encompass methodologies that combine multiple data sources, technologies, and analytical techniques to provide a comprehensive understanding of pollutant dynamics and their impacts on human health and the environment. These approaches aim to overcome the limitations of individual monitoring methods by integrating diverse datasets and tools, thereby enhancing the accuracy, spatial coverage, and predictive capabilities of air quality assessments.

One significant aspect of integrative approaches is the fusion of data from different sensor technologies. For example, combining data from traditional ground-based monitoring stations with satellite remote sensing enables spatially extensive coverage and continuous monitoring of air quality parameters such as PM concentrations (Hu et al., 2011). Satellite data provides valuable information on pollutant transport and spatial variability, complementing ground-based measurements that offer high temporal resolution and local specificity (Hu et al., 2011; Martin, 2008).

Furthermore, integrating meteorological data plays a crucial role in understanding atmospheric processes and their influence on pollutant dispersion. Meteorological variables such as wind speed, temperature, and humidity affect the transport, transformation, and deposition of pollutants in the atmosphere (Li et al., 2017). By incorporating meteorological observations into air quality models, researchers can improve the accuracy of pollutant concentration predictions and assess the potential health impacts more effectively (Aldaweesh, 2019).

Another integral component of integrative approaches is the utilization of advanced modeling techniques, including machine learning and numerical simulation models. Machine learning models, such as neural networks and ensemble methods, leverage complex relationships between environmental variables and air quality parameters to

enhance predictive accuracy (Li et al., 2021). Numerical simulation models, such as Computational Fluid Dynamics (CFD) and Chemical Transport Models (CTMs), simulate pollutant dispersion and transformation processes in the atmosphere, providing insights into pollutant pathways and source-receptor relationships (Pantusheva et al., 2022).

Moreover, the integration of citizen science and participatory monitoring initiatives is increasingly recognized as a valuable approach in air quality monitoring. Citizen scientists equipped with low-cost sensors contribute real-time data collection across diverse geographic areas, enhancing spatial coverage and community engagement in air quality management (Constant, 2018). Integrating these community-based observations with official monitoring networks can improve the spatial resolution of air quality maps and provide localized insights into pollution hotspots and exposure risks (Mahajan et al., 2020).

However, a notable shortcoming of current integrative approaches is their limited incorporation of distance-based feature engineering in air quality modeling. While these approaches integrate meteorological parameters, land use data, and emission inventories, they often overlook the explicit inclusion of spatial distances between monitoring stations as a predictive feature. The omission of distance-based features may lead to suboptimal spatial representation of pollutant concentrations, especially in heterogeneous urban environments where pollutant dispersion can vary widely over short distances as reported by (Aldaweesh, 2019).

Moreover, despite their integrative nature, current approaches still predominantly rely on single-point instantaneous readings of particulate matter (PM), which may not capture the spatial variability of pollutants adequately. This reliance on point measurements limits the ability to assess pollutant gradients and localized hotspots accurately, which

are crucial for identifying vulnerable populations and implementing targeted mitigation strategies.

Therefore, future integrative approaches should prioritize the inclusion of distance-based feature engineering in air quality modeling frameworks. By incorporating spatial distances between monitoring stations as predictive variables, these approaches can better capture the spatial heterogeneity of pollutant concentrations and improve the accuracy of predictive models. Additionally, efforts to enhance the spatial resolution of monitoring networks and integrate real-time data streams from multiple sensors could facilitate more detailed and dynamic assessments of air quality across diverse geographic areas.

### 2.8 Summary and Research Gaps

This chapter has explored key aspects of air quality monitoring, focusing on particulate matter (PM) and its health implications, environmental parameters affecting air pollution dynamics, spatial distribution inference techniques, sensor technology advancements, machine learning applications, and integrative monitoring approaches. The review highlighted significant advancements in sensor technology, such as the PMS5003 sensor, and their ability to measure PM concentrations with varying precision across different particle sizes. Additionally, the review addressed the role of environmental parameters like wind speed, temperature, and humidity in influencing pollutant dispersion and chemical reactions in the atmosphere.

Research gaps identified throughout this review underscore several areas where current methodologies and technologies fall short. One notable gap lies in the limited exploration of how the spatial distance between monitoring stations impacts the accuracy of air quality models. While studies have incorporated meteorological and geographical

features, there remains a lack of research on how including distance-based features and weather data during model training could enhance predictive performance.

Furthermore, the review identified a gap in understanding the relationship between machine learning model feature importance and the separation distances between monitoring stations. Exploring these relationships could provide insights into which environmental factors, influenced by distance, most significantly affect PM concentrations, thus guiding the development of more effective monitoring and prediction models.

Moreover, while integrative approaches combining ground-based measurements, satellite data, and citizen science initiatives show promise, there remains a need for standardized methodologies to integrate diverse datasets seamlessly. Addressing this gap could lead to more robust and comprehensive air quality assessments that account for spatial and temporal variations across different regions.

Additionally, the review highlighted the challenges and limitations of current sensor technologies, including variability in accuracy and reliability, especially concerning low-cost sensors. Addressing these limitations through advancements in sensor calibration, data validation techniques, and quality assurance protocols is crucial for improving the reliability and trustworthiness of air quality data collected from diverse sources.

# CHAPTER THREE: METHODOLOGY

## 3.1 Introduction

This section details the methodology employed for predicting the spatial distribution of particulate matter (PM) concentrations. It encompasses the study site description, a description of the used PM sensor and Weather data components, data collection and preparation, preprocessing, machine learning (ML) model exploration, and evaluation.

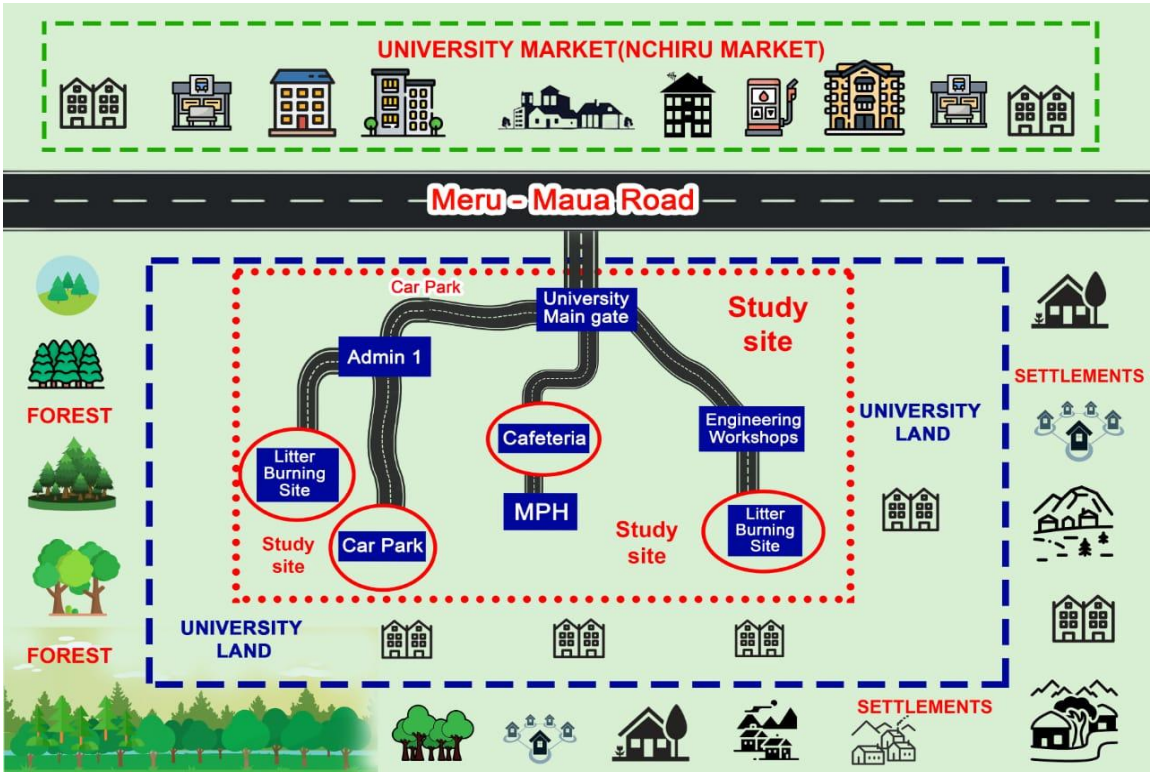
## 3.2 The Study Site

The present work was conducted at Meru University of Science and Technology (MUST) in Kenya. MUST occupies a 600-acre expanse adjacent to a small university town. The campus is bounded by a moderately trafficked road on one side and a tropical rainforest on the opposite side. Human settlement abuts the remaining edges of the university grounds. Highway traffic and other human activities from market, university and adjacent human settlement are likely sources of PM that were transported to the research site which was located within the university grounds. Within the university itself, several factors contribute to PM generation including the campus cafeteria, which caters for a large student population and staff and utilizes firewood as one source of cooking fuel.

The sketch map in Figure 3.1 illustrates the study site location within the university environment relative to the nearby highway, surrounding human settlements, and adjacent forest area.

**Figure 3.1**

*Sketch map of the Study Site-Meru University of Science and Technology.*




*Source: Researcher, 2024.*

### 3.3 Sensor System Design and PM Concentration Data Collection Process

The custom-designed sensor system for measuring PM concentration utilized the Internet of Things (IoT) to log and transmit data to a web server. This system consisted of several components. The core was an ESP32 microcontroller programmed using Arduino sketch to control the system logic, including data acquisition and transmission. For PM concentration measurement, an Optical Particle Counter (PMS5003) was interfaced with the ESP32 to log PM data in three categories namely PM1.0, PM2.5 and PM10. Additionally, a GPS module (NEO-6M) captured spatial information in the form of latitude and longitude. Finally, a weather station kit (SEN-15901) provided weather parameters consisting of wind speed, temperature, and humidity. A GPS module (NEO-

6M) was integrated into the system to log GPS coordinates. The collected data was then transmitted to a designated web server for storage and further analysis using a wireless GSM and Router.

### 3.3.1 PMS5003 sensor principles and operation

The PMS5003 sensor is an optical particulate matter (PM) sensor designed to measure airborne PM concentrations across different size ranges, including PM10 (coarse particles), PM2.5 (fine particles), and PM1.0 (ultrafine particles) (Dubey et al., 2022; Khreis et al., 2022). The sensor operates based on the principle of light scattering, primarily Mie scattering, which occurs when the diameter of the airborne particles is comparable to or larger than the wavelength of the incident light (Ishimaru, 1997; Hergert and Wriedt, 2012).

The PMS5003 contains a laser diode that emits a coherent and collimated beam of light into a detection chamber. When airborne particles pass through the illuminated region, the incident light interacts with these particles, leading to scattering in different directions. A photodetector positioned at a specific angle measures the intensity of the scattered light, and this intensity is used to infer particle concentration and size distribution (Alfano et al., 2020b; Dubey et al., 2022).

**Mie Scattering and Particle Measurement.** The PMS5003 sensor relies on Mie theory to model how light interacts with spherical particles in the detection chamber. Unlike Rayleigh scattering, which is dominant for particles much smaller than the wavelength of light, Mie scattering accounts for particles with diameters comparable to or larger than the laser wavelength (typically in the range of 600–800 nm) (Meyer-Arendt, 1989). Mie scattering is dependent on several factors, including:

i. The particle radius $(r)$.

ii. The refractive index $(m)$ of the particle relative to the surrounding medium.

iii. The wavelength $(\lambda)$ of the incident laser beam.

iv. The scattering angle $(\theta)$ at which light is detected.

The scattering intensity is determined using the Mie scattering coefficient, which is derived from Maxwell's equations and depends on the particle size parameter x, defined as: $X=\frac{2\pi r}{\lambda}$, where $r$ is the particle radius and $\lambda$ is the laser wavelength.

For a single spherical particle, the intensity of scattered light at a given angle $\theta$ is given by:

$$I(\theta) = I_0.\frac{1}{r^2}.S_1^2(\theta) + S_2^2(\theta) \quad (3.1)$$

where $S_1(\theta)$ and $S_1(\theta)$ are the Mie scattering amplitude functions dependent on the refractive index and size parameter.

**Single-Scattering Approximation in the PMS5003.** For simplification, the PMS5003 sensor is typically modeled under the single-scattering assumption, where each particle independently scatters light without multiple scattering events occurring. This assumption holds when particle concentrations are low enough that secondary interactions between particles and scattered photons are negligible (Ishimaru, 1997). Under this condition, the total scattered intensity measured by the photodetector is the sum of contributions from individual particles.

To estimate particle concentration, the PMS5003 applies a calibration curve based on empirical data collected from known aerosol samples. The intensity of the detected scattered light is processed using signal amplification and computational algorithms that classify particles based on their scattering characteristics.

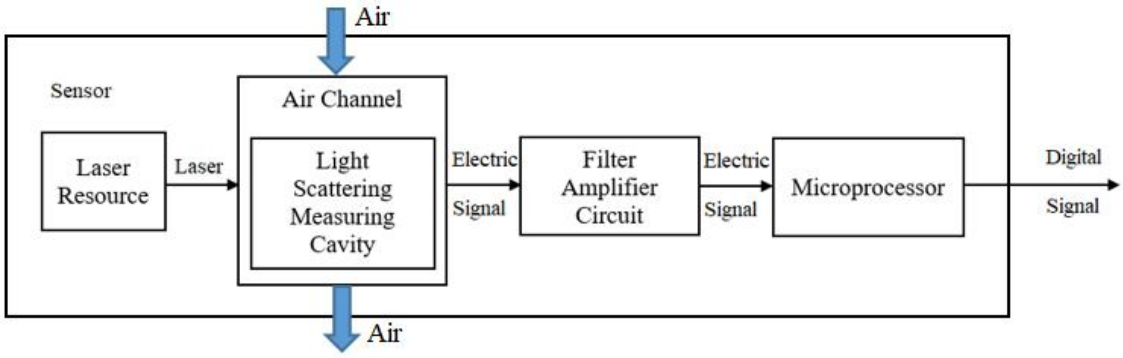
**Comparison with LIDAR-Based Scattering.** While Mie theory is also used in LIDAR applications, the PMS5003 functions as a miniaturized low-power optical scattering sensor rather than an active remote sensing system. Unlike LIDAR, which typically

requires complex backscattering equations and range-dependent corrections, the PMS5003 sensor measures scattering at a fixed angle, assuming a controlled detection volume.

Figure 3.2 (Modified from (PMS5003 Series Manual by Adafruit Industries LLC Datasheet | DigiKey, 2024)) illustrates the PMS5003 signal processing flow, which includes light scattering detection, photodiode signal processing, analog-to-digital conversion, and final particle concentration estimation based on embedded algorithms.

**Figure 3.2**

*Flow of laser scattering in PMS5003 Sensor.*




*Source: Researcher, 2024.*

As depicted in Figure 3.2, a laser light source interacts with particles passing through its detection chamber. The scattered light, which is a direct result of this interaction, is detected by a photodetector within the detection cavity. The scattered light is then converted to electrical signal. This electrical signal is subsequently sent to an amplifier which amplifies it for further processing within the microprocessor, employing the described Mie Theory algorithm. In analyzing the light scattering patterns, the sensor can differentiate between different-sized particles, with larger particle diameters contributing more to the scattered light intensity. This enables the sensor to distinguish between coarse (PM10), fine (PM2.5), and ultrafine (PM1.0) particles based on their respective

size ranges. The PMS5003 sensor also incorporates a built-in fan that helps to maintain a continuous and consistent airflow through the detection chamber, ensuring reliable measurements (Alfano et al., 2020b). The sensor provides real-time data on PM concentrations, allowing for immediate monitoring and analysis of air quality conditions.

### 3.3.2 Design of the cloud-based server

On the cloud-based server, a PHP-MySQL database was implemented for data storage and management. This database logged environmental data (PM1.0, PM2.5 PM10, Latitude and Longitude) collected from the sensor modules. The server side application also implemented Google Sheets integration to enable real-time data access and visualization over the internet.

### 3.3.3 Data collection process

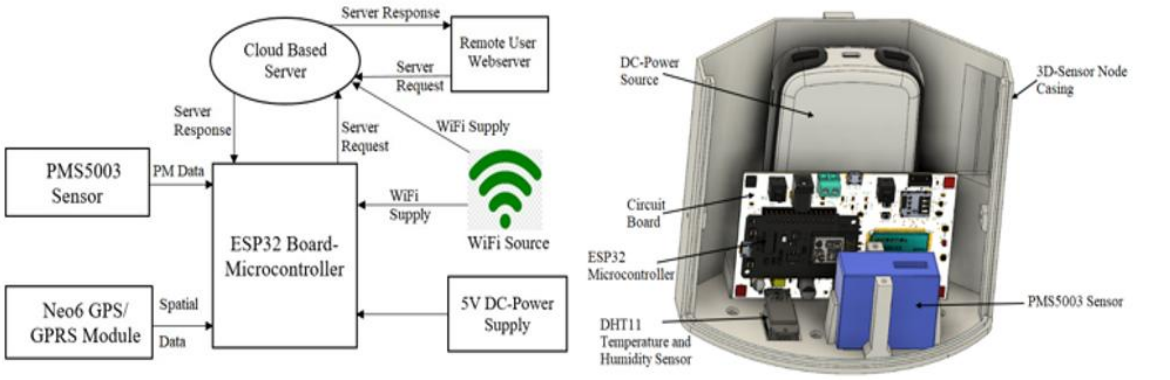
The data collection process employed a two-part strategy. A stationary central sensor node (S1) continuously recorded PM concentrations (PM1.0, PM2.5, PM10) and its location (latitude and longitude) for three weeks, establishing a baseline. Alongside S1, an IoT-based weather monitoring system was deployed to capture meteorological parameters essential for the study.

The Weather Station Kit (SEN-15901) was used for weather data collection, measuring wind speed, temperature, and humidity in real time. This kit integrates multiple sensors capable of recording temperature, humidity, rainfall, wind speed, and air pressure. Sensor readings were wirelessly transmitted to a receiver unit, which then relayed the data to a PC via a USB connection. The PC software processed and stored the incoming data, enabling continuous monitoring and analysis of weather conditions. This integrated approach ensured synchronized environmental and air quality data collection, enhancing the study's ability to assess PM concentration variations in relation to meteorological factors.

To investigate spatial variations in PM concentrations, a separate mobile sensor node (S3) was moved randomly within the vicinity of S1, collecting the similar data as S1. This dual approach allowed for real-time data visualization through a cloud-based server. Figure 3.3 and Figure 3.4 illustrates the system architecture and real-time data visualization frameworks, respectively. Specifically, Figure 3.4(a) depicts a sample display for the central sensor node (S1), showcasing sample measurements for PM across the three scales (PM1.0, PM2.5, PM10) alongside its corresponding location. Similarly, Figure 3.4(b) displays real-time data for the mobile sensor node (S3), including PM concentrations and spatial coordinates. The data displays clearly differentiate between the two sensor nodes (S1 and S3).

**Figure 3.3**

*Illustration of IoT enabled sensor system for measuring PM* concentrations.




*(a) Block representation of the IOT-Based PM Sensor Node*

*(b) Top View of the IoT-Based Sensor Node Components*

*Source: Researcher, 2024.*

**Figure 3.4**

*Illustration of the IoT enabled sensor system user dashboards.*

**Particulate Pollutants Research**

*PARTICULATE_CENTER_SENSOR_NODE_DATASET*

| ID | PM1_0_S1 | PM2_5_S1 | PM10_S1 | Latitude_S1 | Longitude_S1 | Timestamp |
|---|---|---|---|---|---|---|
| 80440 | 2 | 3 | 3 | 0.13414299 | 37.7092895 | 2024-04-06 14:49:22 |
| 80439 | 2 | 3 | 3 | 0.13414299 | 37.7092895 | 2024-04-06 14:49:17 |
| 80438 | 2 | 3 | 3 | 0.13413700 | 37.7092933 | 2024-04-06 14:49:13 |
| 80437 | 1 | 2 | 2 | 0.13413299 | 37.7092933 | 2024-04-06 14:49:10 |

*(a) Central Sensor Node Real-Time PM measurement*

**Particulate Pollutants Research**

*PARTICULATE_POLLUTANTS_MOBILE_SENSOR_NODE_DATASET*

| ID | PM1_0_S3 | PM2_5_S3 | PM10_S3 | Latitude_S3 | Longitude_S3 | Timestamp |
|---|---|---|---|---|---|---|
| 74183 | 3 | 6 | 7 | 0.13678799 | 37.7104148 | 2024-04-06 13:30:28 |
| 74182 | 3 | 6 | 7 | 0.13678400 | 37.7104187 | 2024-04-06 13:30:15 |
| 74181 | 1 | 1 | 3 | 0.13678300 | 37.7104187 | 2024-04-06 13:30:02 |
| 74180 | 1 | 1 | 3 | 0.13678899 | 37.7104187 | 2024-04-06 13:29:49 |

*(b) Mobile Sensor Node Real-Time PM measurement*

*Source: Researcher, 2024.*

Figure 3.3(a) illustrates the system architecture. Figure 3.3(b) focuses on top view of the hardware components within a single sensor node. Figure 3.4(a) and Figure 3.4(b), depict real-time data visualizations for the stationary and mobile sensor nodes, respectively.

### 3.3.4 Data set preparation and preprocessing

The first preprocessing step involved calculating the separation distances between mobile and central sensor nodes. This spatial information is vital for understanding how PM concentrations vary across the sensor network. To compute the separation distance, the Euclidean distance formula was employed, providing a straightforward and efficient method for determining the straight-line distance between each mobile sensor and the central node based on their latitude and longitude coordinates. The Euclidean distance is a well-known mathematical approach used in various studies related to spatial data analysis (Behrens et al., 2018; Jones et al., 2010), offering a clear and direct measure of the spatial separation between two points on a flat surface.

Using the Cartesian coordinates $(x_1, y_1)$ and $(x_2, y_2)$ of the central and mobile sensor nodes, respectively, the Euclidean distance is computed using the distance formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.2}$$

Where:

i. $x_1$ and $y_1$ represent the latitude and longitude of the central sensor node.

ii. $x_2$ and $y_2$ represent the latitude and longitude of the mobile sensor node.

The calculated distance in radians is converted to meters by multiplying it by the Earth's radius, approximated as $R = 6{,}371{,}000$ meters:

$$Euclidean\_D = d \times R \tag{3.3}$$

Equation 3.2 was applied as follows:

First, latitude and longitude values were initially measured in degrees so they were converted to radians, as trigonometric functions in Python operate on radians. The conversion was performed using the following formulation:

$$rad = deg \times \frac{\pi}{180} \tag{3.4}$$

For latitude (denoted here as $lat$) and longitude (denoted here as $lon$):

$$\left.\begin{aligned} lat_{rad} &= lat \times \frac{\pi}{180} \\ lon_{rad} &= lon \times \frac{\pi}{180} \end{aligned}\right\} \tag{3.5}$$

The orientation or angle $\theta$ between two points is calculated using the arctangent function, providing the direction of the line connecting the two points relative to the x-axis:

$$\theta = tan^{-} \left(\frac{(y_2 - y_1)}{(x_2 - x_1)}\right) \tag{3.6}$$

This angle $\theta$ indicates the orientation of the line segment connecting the central and mobile sensors.

Here, $y_2 - y_1$ represents the change in latitude, and $x_2 - x_1$ represents the change in longitude.

The angle $\theta$ is then mapped to one of the cardinal directions (East, North, South, West), providing the mobile sensor node orientation relative to the central sensor node.

Transforming the computed Euclidean distance (referred through subsequent sections as the transformed Euclidean distance) from degrees to meters using the equation 3.3 was necessary to ensure accurate spatial measurements for the analysis. Since latitude and longitude coordinates are typically represented in degrees, converting them to meters provides a more meaningful and standardized measure of separation distance, facilitating precise spatial analysis and comparison across different instances.

**Assumption (On Euclidean Distance).** Given the relatively short distances between the central and mobile sensor nodes (1-meter intervals), Euclidean distance was deemed appropriate for this study. The small-scale spatial separation within the sensor network justifies the assumption of a flat Earth, a fundamental premise of Euclidean geometry. Over short distances, the Earth's curvature is negligible, allowing latitude and longitude coordinates to be approximated as points on a Cartesian plane. This approach is consistent with previous research utilizing Euclidean geometry for local spatial analysis. The progressive 1-meter intervals between sensor nodes further minimize any potential distortion in distance measurements, ensuring accurate spatial analysis of PM concentrations.

Another preprocessing step focused on incorporating sensor orientation data. Label encoding was then applied to convert the categorical sensor node orientation data into numerical representations suitable for model training.

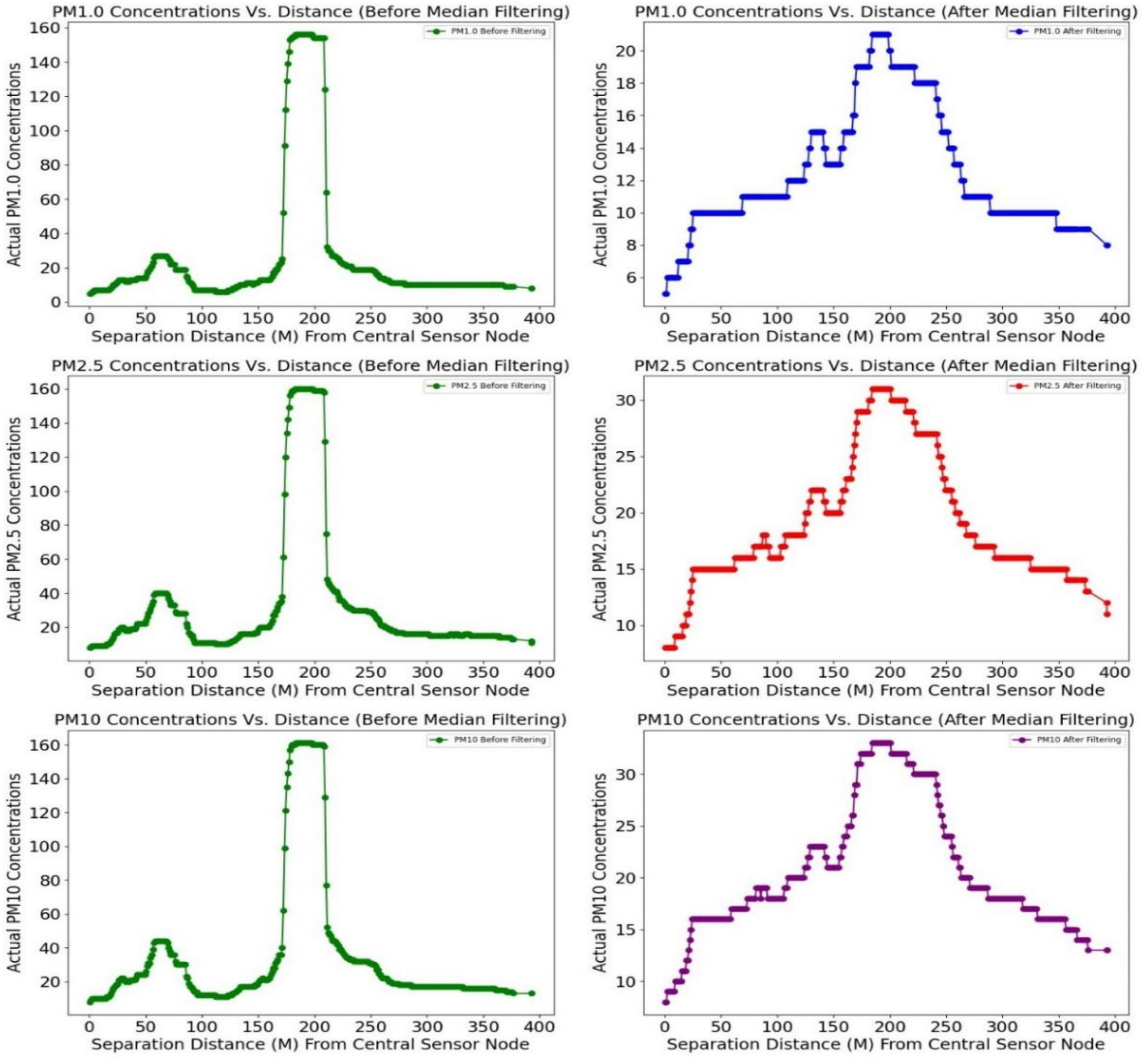
Furthermore, to mitigate the impact of outliers (noise) in the raw data that could distort model performance, median filtering was employed. This technique is a common

practice in data preprocessing, as demonstrated in studies focused on air quality data analysis (Cofre-Martel et al., 2021; Ganguli, 2002). Figure 3.4 shows the results of preprocessed PM1.0, PM2.5 and PM10 features based on the application of median filtering to remove outliers across the PM data.

Finally, data scaling was performed to normalize the feature values within the dataset. This step prevents features with larger scales from disproportionately influencing the model training process. The preprocessed data produced a well-structured feature matrix, comprising 75,999 rows per feature, with dependent variables (target features) including the three PM scales (PM1.0, PM2.5, and PM10 concentrations). The independent variables (input features) include wind speed, temperature, humidity, Euclidean_D between sensors, and the encoded sensor orientations. This prepared dataset served as the foundation for training and evaluating the performance of the four machine learning models explored in this study.

**Figure 3.5**

*Application of Median Filtering to Remove Spurious Values in the Raw Data.*




*Source: Researcher, 2024.*

### 3.4 Training and Evaluation Strategy

The study evaluated the performance of four machine learning models for predicting PM concentrations: Long Short-Term Memory (LSTM) networks, Artificial Neural Networks (ANNs), Support Vector Regression (SVR), and Random Forest (RF) models. LSTMs were chosen due to their ability to learn from sequential data, making them particularly suitable for analyzing time-series data of PM concentrations, as evidenced in

(Batur Şahin & Diri, 2019; Denny et al., 2018). ANNs were included due to their adaptability and capacity to handle complex relationships within the data, as discussed in (Pawul and Śliwka, 2016; Zhang, 2018). SVR models were selected for their effectiveness in modeling non-linear relationships, as demonstrated in (Kshirsagar and Khare, 2023), while Random Forests were chosen for their robustness and ability to handle high-dimensional data, as highlighted in (Liu et al., 2012).

The training of machine learning models was conducted using multiple feature combinations derived from the collected and preprocessed data. Specifically, models were trained using the following feature sets: (i) 'Euclidean_D' and 'Orientation'; (ii) 'Wind speed', 'Euclidean_D', and 'Orientation'; (iii) 'Temperature', 'Euclidean_D', and 'Orientation'; (iv) 'Humidity', 'Euclidean_D', and 'Orientation'; (v) 'Temperature', 'Humidity', 'Euclidean_D', and 'Orientation'; (vi) 'Wind speed', 'Temperature', 'Euclidean_D', and 'Orientation'; (vii) 'Wind speed', 'Humidity', 'Euclidean_D', and 'Orientation'; and (viii) 'Wind speed', 'Temperature', 'Humidity', 'Euclidean_D', and 'Orientation'. These feature combinations aimed to explore the influence of different environmental factors on model performance, ensuring a comprehensive assessment of their predictive capacity.

The training and evaluation strategy involved splitting the dataset into three categories: training, validation, and testing sets. Each machine learning model was trained using specific hyperparameters and evaluated using various performance metrics, including training and validation R-squared ($R^2$) scores. The $R^2$ score signifies the proportion of variance in the dependent variable (PM concentration) explained by the independent variables (the features used for prediction) (Di Bucchianico, 2007). In this context, a higher $R^2$ score signifies a better model fit and stronger predictive capability. $R^2$ scores

were computed using the formula provided in Equation 3.7, used also by (Han et al., 2021; Kshirsagar and Khare, 2023; Sakhrieh et al., 2021).

$$R^2 = 1 - \left( \frac{\sum (y_{actual} - y_{predicted})^2}{\sum (y_{actual} - y_{mean})^2} \right) \quad (3.7)$$

Variables;

- $y_{actual}$. $Actual\ PM\ concentration\ values$
- $y_{predicted}$. $Predicted\ PM\ concentration\ values\ by\ the\ model$
- $y_{mean}$. $Mean\ of\ the\ actual\ PM\ concentration\ values$

Additionally, RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) metrics were applied as evaluation metrics to provide a comprehensive assessment of model performance. These metrics were chosen in combination with $R^2$ scores because they offer different insights into the model's predictive accuracy and error characteristics, making them valuable for a robust evaluation.

RMSE measures the square root of the average squared differences between the predicted and actual values, providing an indication of the model's prediction error magnitude. It is particularly sensitive to large errors, making it useful for identifying models that perform poorly on certain subsets of data. The RMSE is defined as provided in Equation 3.8.

$$RMSE = \sqrt{\frac{\sum (y_{actual} - y_{predicted})^2}{n}} \quad (3.8)$$

Variables:

- $y_{actual}$: $Actual\ PM\ concentration\ values$
- $y_{predicted}$: $Predicted\ PM\ concentration\ values\ by\ the\ model$
- $n$: $Number\ of\ data\ points\ (samples)$

MAE measures the average absolute differences between the predicted and actual values, providing a straightforward measure of prediction accuracy. Unlike RMSE, MAE is less sensitive to large errors, offering a balanced view of model performance across the dataset. The MAE is defined as according to Equation 3.9.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \left| y_{actual} - y_{predicted} \right| \quad (3.9)$$

Variables:

- $y_{actual}$: $Actual\ PM\ concentration\ values$.
- $y_{predicted}$: $Predicted\ PM\ concentration\ values\ by\ the\ model$.
- $n$: $Number\ of\ data\ points\ (samples)$.

This study assessed the performance of four machine learning models in predicting particulate matter (PM) concentrations. The models were trained on data collected from described IoT-based sensor nodes. The data included PM concentrations (PM1.0, PM2.5, and PM10) along with various influencing factors: weather data (wind speed, temperature, and humidity) and geometric features (Euclidean_D and Orientation)

An Artificial Neural Network (ANN) with a Multilayer Perceptron (MLP) architecture was implemented. The MLP consisted of multiple densely connected layers with ReLU activation functions in hidden layers and linear activation in the output layer. The model was trained using mean squared error (MSE) loss function and the Adam optimizer to minimize the difference between predicted and actual PM values.

A Long-Short Term Memory (LSTM) network was employed to capture temporal patterns. The LSTM architecture consisted of stacked LSTM layers followed by dropout layers to prevent overfitting and a dense output layer. The model was trained using MSE loss and Adam optimizer to learn the mapping between sequences of input features and corresponding PM concentrations.

Support Vector Regression (SVR) was employed to find a hyperplane that best approximates the relationship between input variables and PM concentrations. SVR utilizes an epsilon-insensitive loss function, which ignores errors below a certain threshold. It also uses kernel functions to map data into higher dimensions for improved separability. The specific SVR model used a radial basis function (RBF) kernel and was tuned with appropriate hyperparameters, including the C (penalty) parameter and the epsilon (insensitivity) parameter.

Finally, a Random Forest Regressor was implemented. This ensemble learning method constructs multiple decision trees. Each tree is trained on a random subset of the data and selects a random subset of features at each split. The final prediction is the average of predictions from all trees, reducing variance and improving generalization performance. Randomness is introduced by selecting random subsets of features and using bootstrap aggregating (bagging) during training. The model is trained by recursively growing trees based on a chosen criterion (mean squared error in this case) until a stopping criterion is met. The final output is obtained by averaging the predictions from all trees.

The best performing model was further analyzed to assess its prediction accuracy and identify any potential trends. To quantify the improvement achieved by including weather data, a signal energy analysis was employed. This analysis calculates the cumulative energy contained within the concentration values for both the actual PM concentrations measurements and the model's predictions. The signal energy analysis was performed according to the following framework.

### 3.4.1 Signal energy analysis

Consider the Ohms Law $V = IR$, where $V$ is the voltage, $I$ is the current, and $R$ is the resistance. The Power ($P$) in terms of Ohm's Law can be expressed as;

$$P = IV = I^2R = \frac{V^2}{R} \tag{3.10}$$

Therefore, Signal Energy (E) over some spatial domain $T$:

$$E = \int_0^T P(x)dx = \int_0^T V(x)I(x)dx$$

In the context of predictions, this work considered the actual and predicted values as signals.

Signal Energy (E) for a discrete signal $y$ of length $N$:

$$E_N = \sum_{i=1}^{N} y_i^2 \tag{3.11}$$

Using Equation 3.11 in the analysis:

Actual Signal Energy ($E_{actual}$ ): $E_{actual} = \sum_{i=1}^{N} |y_{actual,i}|^2$

Predicted Signal Energy ($E_{predicted}$ ): $E_{predicted} = \sum_{i=1}^{N} |y_{predicted,i}|^2$

Signal Energy Error: $E_{error} = \sum_{i=1}^{N} |y_{actual,i} - y_{predicted,i}|^2$

Then, the relative percentage error is computed according to equation 3.12.

$$R_{Percentage\ Eror} = \left(\frac{E_{error}}{E_{actual}}\right) \times 100\% \tag{3.12}$$

Interpretation: As an example, a percentage error of $3\%$ in signal energy represents the difference between the actual signal energy and the predicted signal energy as a proportion of the actual signal energy.

Accuracy Indicator: A $3\%$ error indicates that the predicted signal energy deviates from the actual signal energy by only $3\%$. This suggests that the prediction model is highly accurate, capturing the majority of the signal's characteristics correctly.

Performance Evaluation: In practical terms, this small error margin demonstrates that the model's predictions are very close to the real measurements. Such a low percentage error implies that the model is reliable and its predictions can be trusted for practical applications, such as monitoring air quality.

Model Robustness: A $3\%$ error reflects the robustness of the model in handling various input conditions and still providing precise predictions. It indicates that the model's performance is stable and consistent across different test scenarios.

Practical Example: Suppose the actual signal energy ($E_{actual}$) is $1000units$. A $3\%$ error would mean the predicted signal energy ($E_{predicted}$) is either $970units$ or $1030units$. The difference ($30units$) represents a small fraction of the total energy, hence reflecting high prediction fidelity.

Therefore, a lower signal energy error, obtained by summing the squared differences between these values, indicates a better match between the predicted and actual PM concentrations. This error is then translated to a percentage error, providing a quantitative measure of the model's accuracy. Following this initial assessment, a moving average technique was applied to analyze trends in prediction accuracy across the 160-meter radius. The moving average smooths a signal by replacing each data point with the average of neighboring points within a specified window size. The mathematical formula for computing the moving average at each point $y_i$ in the signal can be represented according to equation 3.13, (Guiñón et al., 2007; Hyndman, 2010):

$$Smoothed(i) = \frac{1}{N}\sum_{j=i-k/2}^{i+k/2} Original(j) \quad (3.13)$$

Where:

- $Original(j)$ is the original accuracy value at index $j$.
- $Smoothed(i)$ is the smoothed accuracy value at index $i$.
- $N$ is the total number of points in the window (window size).
- $k$ is the half-width of the window (number of points on each side of the current point).

Equation 3.19 calculates the average of the original accuracy values within a window centered at each index $i$, effectively smoothing out the curve.

By smoothing out fluctuations in the data, this approach provides a clearer picture of how accuracy varies with distance from the central sensor node.

# CHAPTER FOUR: RESULTS AND DISCUSSION

## 4.1 Introduction

This chapter presents the analysis and discussion of four machine learning models that were experimented, including: Artificial Neural Networks (ANNs), Support Vector Regression (SVR), Long Short-Term Memory (LSTM), and Random Forest. The analysis focuses on the model's ability to predict the spatial distribution of PM1.0, PM2.5, and PM10 concentrations, both before and after deployment. Pre-deployment analysis evaluates the models based on three chosen metrics: $R^2$ scores, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and average prediction accuracy. Additionally, this section explores the impact of feature importance ranking on model performance and identifies the best performing model. Post-deployment analysis focuses on the selected best model. It examines the instantaneous alignment between actual and predicted PM concentrations. Further model validation is achieved through signal energy error analysis, along with a 1D moving average signal energy analysis.

## 4.2 Pre-Deployment Analysis

ANN, LSTM, SVR and RF models were trained to predict the PM values at a given distance from the reference point (the reference point will be referred as the Point of Interest (PoI)) given the inputs of the model being the PM value at the reference point and geometric features. Geometric features entailed transformed Euclidean distance (Euclidean_D) and Orientation of the PoI with respect to the reference point. Separately, the models were trained to predict the PM value at the PoI but with the inputs incorporating the PM values at the reference point, geometric features (Euclidean_D and Orientation) and weather features (Wind speed, Temperature and Humidity). Weather data consisted of wind speed, temperature and humidity. The performance of the machine learning models were evaluated in terms of the attained $R^2$ scores and model

Root Mean Squared (RMS) errors and Mean Absolute Errors (MAE). The training and evaluation performance of the four ML models is presented in Table 4.1 to Table 4.10.

### 4.2.1 Machine leaning models training and validation for PM1.0

**Table 4.1**

*Evaluation of Machine Learning Models for PM1.0 Prediction.*

| Model | Feature Combinations | Train $R^2$Score | Train RMSE | Train MAE | Validation $R^2$Score | Validation RMSE | Validation MAE |
|---|---|---|---|---|---|---|---|
| ANN | Euclidean_D, Orientation | 0.9746 | 2.7696 | 1.0535 | 0.9711 | 1.8719 | 0.8125 |
| | Wind speed, Temperature, Euclidean_D, Orientation | 0.9790 | 2.5225 | 0.8002 | 0.9756 | 1.7201 | 0.7952 |
| SVR | Euclidean_D, Orientation | 0.9787 | 2.7479 | 1.2504 | 0.9648 | 2.0516 | 1.1397 |
| | Wind speed, Euclidean_D, Orientation | 0.9792 | 2.7150 | 1.2304 | 0.9679 | 1.9590 | 1.0621 |
| LSTM | Euclidean_D, Orientation | 0.9856 | 2.0778 | 0.8632 | 0.9694 | 1.9154 | 1.1286 |
| | Temperature, Euclidean_D, Orientation | 0.9841 | 2.1823 | 0.8946 | 0.9578 | 2.2482 | 1.2846 |
| Random Forest | Euclidean_D, Orientation | 0.5878 | 1.7222 | 1.3607 | 0.5472 | 1.8086 | 1.4229 |
| | Temperature, Humidity, Euclidean_D, Orientation | 0.6486 | 1.5902 | 1.2411 | 0.5354 | 1.8320 | 1.4314 |

*Source: Research data, 2024.*

Table 4.1 illustrates the impact of including weather parameters on the performance of various machine learning models predicting PM concentrations up to 160 meters spatially. The results clearly demonstrate that incorporating weather data significantly improves the models' ability to capture the spatial variability of PM concentrations.

Initially trained without weather parameters, the ANN achieved a validation $R^2$ of 0.9711, indicating a strong fit but with some room for improvement. However, incorporating wind speed and temperature into the training data boosted the validation $R^2$ to 0.9756, suggesting a closer match between predicted and actual PM values. This improvement is further reflected in the reduced RMSE (1.8719 to 1.7201) and MAE (0.8125 to 0.7952), signifying a decrease in the average prediction errors. The results suggest that the model's predictions deviated from actual PM1.0 concentrations by only 0.7952 μg/m³ on average, which is significant given the PM1.0 range of 5 to 25.

Similarly, the LSTM model also benefited from the inclusion of weather data. Without them, its validation $R^2$ stood at 0.9694, while the RMSE and MAE were 1.9154 and 1.1286, respectively. After incorporating temperature, the validation $R^2$ slightly dropped with the highest validation value at 0.9578, indicating a slightly better fit. The RMSE and MAE also increased to 2.2482 and 1.2846, respectively, suggesting that inclusion of weather parameters in the model training did not significantly improve LSTM generalization on unseen data. Further, this result implies that the model's predictions deviated from actual PM1.0 concentrations by only 1.2846 μg/m³ on average.

The SVR model also exhibited enhanced performance when trained with additional weather parameters. Initially, its validation $R^2$ was 0.9648, while the RMSE and MAE were 2.0516 and 1.1397, respectively. After including wind speed and temperature, the validation $R^2$ increased to 0.9679, suggesting a slightly improved fit. The RMSE and MAE also decreased to 1.9590 and 1.0621, respectively, indicating a reduction in average prediction errors. This implies that the model's predictions deviated from actual PM1.0 concentrations by only 1.0621 μg/m³ on average.

Conversely, the Random Forest model consistently showed lower performance compared to the other models, regardless of the feature combinations used. Even with additional

weather parameters, its validation $R^2$ remained relatively low at 0.5354, while the RMSE and MAE were 1.8320 and 1.4314, respectively. This suggests that the model's predictions deviated from actual PM1.0 concentrations by an average of 1.4314 µg/m³, which is significantly higher compared to the other models.

Among the various impactful feature combinations, the results indicate that the best-performing model is the ANN. This model achieved the highest validation $R^2$ scores (0.9756) and the lowest RMSE (1.7201) and MAE (0.7952) values, indicating its effectiveness in predicting PM concentrations spatially. The inclusion of wind speed and temperature as weather parameters significantly improved the model's performance, particularly for the ANN and LSTM models, showcasing the importance of these parameters in accurate PM concentration prediction.

**4.2.2 Machine leaning models training and validation for PM2.5**

The PM2.5 model training and evaluation analysis in Table 4.2 reveals the varying impacts of feature selection on the performance of different machine learning models. The ANN model demonstrated substantial improvements in its performance metrics when humidity was included as a feature. Initially, with Euclidean_D and Orientation, the ANN achieved a validation $R^2$ score of 0.9692, RMSE of 2.5260, and MAE of 1.1488. However, adding humidity improved the validation $R^2$ to 0.9789, reduced RMSE to 2.2139, and lowered MAE to 1.0012, indicating more accurate predictions with average deviations of 1.0012 µg/m³ from actual PM2.5 concentrations, which ranged between 5 and 33 µg/m³.

The SVR model also benefited from additional features, showing a validation $R^2$ of 0.9737, RMSE of 2.3085, and MAE of 1.1139 with Euclidean_D and Orientation. The inclusion of temperature and humidity further improved its performance to a validation $R^2$ of 0.9751, RMSE of 2.2459, and MAE of 1.0905, suggesting a slight enhancement in

prediction accuracy. However, the LSTM model exhibited poor generalization, with its validation $R^2$ dropping significantly to 0.8684 and its RMSE and MAE increasing to 5.1970 and 4.1140, respectively, when trained with Euclidean_D and Orientation. Including temperature and humidity did not substantially improve these metrics, resulting in a validation $R^2$ of 0.9471, RMSE of 3.2951, and MAE of 2.5240, which indicates considerable prediction errors.

The Random Forest model consistently performed worse than the other models, showing a validation $R^2$ of 0.4873, RMSE of 2.6127, and MAE of 2.0474 with Euclidean_D and Orientation. Adding temperature as a feature did not significantly improve its performance, with a validation $R^2$ of 0.4702, RMSE of 2.6557, and MAE of 2.0712, indicating less accurate predictions compared to ANN and SVR.

The results in Table 4.2 suggest that humidity is a significant factor in predicting PM2.5 concentrations, aligning with other studies that emphasize the role of environmental variables in air quality models (Xu et al., 2021; Li et al., 2020). Among the models, ANN emerged as the most suitable for predicting PM2.5 concentrations, achieving the highest training and validation $R^2$ scores (0.9805 and 0.9789, respectively) and the lowest RMSE (2.2139) and MAE (1.0012), outperforming SVR, LSTM and RandomForest. The LSTM model's poor generalization, evidenced by higher RMSE and MAE, indicates its limitations in predicting PM2.5 concentrations accurately. Similarly, the Random Forest model's performance did not improve significantly with the inclusion of weather parameters, reflecting its limited predictive capability for this task.

**Table 4.2**

*Impact of Feature Selection on PM2.5 Prediction Model Performance.*

| Model | Feature Combinations | Train $R^2$ Score | Train RMSE | Train MAE | Validation $R^2$ Score | Validation RMSE | Validation MAE |
|---|---|---|---|---|---|---|---|
| ANN | Euclidean_D, Orientation | 0.9766 | 3.0048 | 1.3408 | 0.9692 | 2.5260 | 1.1488 |
| | Humidity, Euclidean_D, Orientation | 0.9805 | 2.7428 | 1.0591 | 0.9789 | 2.2139 | 1.0012 |
| SVR | Euclidean_D, Orientation | 0.9814 | 2.8503 | 1.3904 | 0.9737 | 2.3085 | 1.1139 |
| | Temperature, Humidity, Euclidean_D, Orientation | 0.9815 | 2.8409 | 1.3895 | 0.9751 | 2.2459 | 1.0905 |
| LSTM | Euclidean_D, Orientation | 0.9867 | 2.2526 | 1.1252 | 0.8684 | 5.1970 | 4.1140 |
| | Temperature, Humidity, Euclidean_D, Orientation | 0.9752 | 3.0761 | 2.2118 | 0.9471 | 3.2951 | 2.5240 |
| Random Forest | Euclidean_D, Orientation | 0.5224 | 2.5202 | 1.9744 | 0.4873 | 2.6127 | 2.0474 |
| | Temperature, Humidity, Euclidean_D, Orientation | 0.5986 | 2.3101 | 1.8011 | 0.4702 | 2.6557 | 2.0712 |

*Source: Research data, 2024.*

Comparing the results, the inclusion of weather parameters alongside geometric features did not drastically enhance model performance, as evidenced by only slightly improved $R^2$ scores and reduced RMSE and MAE. This observation is consistent with the performance patterns of the LSTM and Random Forest models, where the added weather features did not significantly elevate prediction accuracy. Similar research has shown varying impacts of feature inclusion, often highlighting that while certain environmental variables can improve predictions, their influence may be model-dependent (Abecasis et

al., 2022; Li et al., 2019). Overall, the ANN model's superior performance, validated by its higher accuracy metrics, underscores its robustness in PM2.5 concentration prediction.

### 4.2.3 Machine leaning models training and validation for PM10

Table 4.3 illustrates the impact of feature combinations on the performance of various machine learning models predicting PM10 concentrations. Initially, the ANN model achieved a validation $R^2$ of 0.9775, indicating a strong fit, which was further enhanced to 0.9793 upon incorporating wind speed and temperature. This improvement was accompanied by a decrease in RMSE from 2.231 to 2.143 and MAE from 1.1120 to 1.0037, highlighting the model's enhanced precision. The deviation from actual PM10 concentrations, which ranged between 5 and 35 μg/m³, was minimal, with an average prediction error of 1.0037 μg/m³. This suggests that wind speed and temperature are crucial parameters for improving ANN model performance.

Similarly, the SVR model showed enhanced performance with additional weather parameters, achieving a validation $R^2$ of 0.9739 initially, which increased slightly to 0.9751 with humidity included. The RMSE and MAE also decreased from 2.3492 to 2.2940 and 1.1315 to 1.1140, respectively. These improvements indicate that while the SVR model benefitted from weather data, the enhancements were marginal, with the model's predictions deviating from actual PM10 concentrations by about 1.1140 μg/m³ on average.

In contrast, the LSTM model exhibited poor generalization. Without weather parameters, its validation $R^2$ was 0.9018, with RMSE and MAE at 4.6475 and 3.5176, respectively. Even with additional weather parameters, the validation $R^2$ improved to 0.9602, yet the RMSE and MAE remained relatively high at 2.9566 and 1.8514. These metrics indicate substantial prediction errors, suggesting that LSTM struggled to accurately predict PM10 concentrations, deviating by 1.8514 μg/m³ on average.

The RandomForest model consistently underperformed compared to the other models, with validation $R^2$ values of 0.4672 and 0.4569 for the different feature sets. The RMSE and MAE also remained high, around 2.8025 and 2.2008, respectively, indicating significant prediction deviations. This underscores the model's limited ability to predict PM10 concentrations accurately, with an average error of 2.2008 μg/m³, reflecting a considerable deviation from actual concentrations.

The inclusion of weather parameters did not significantly improve model performance across the board. For ANN, the slight enhancements in $R^2$, RMSE, and MAE suggest some benefit, but for LSTM and RandomForest, the improvements were negligible. This aligns with other studies where the inclusion of environmental data has shown mixed results in improving PM predictions, depending on the model architecture and the specific weather parameters used (Abecasis et al., 2022; Li et al., 2019). The ANN model consistently outperformed SVR and LSTM, with lower RMSE and MAE, validating its suitability for PM10 prediction.

LSTM's poor generalization compared to ANN and SVR is evident, indicating it may not be the best choice for this type of prediction. Similar observations were made for RandomForest, where adding weather parameters did not significantly enhance performance. These findings align with current research, suggesting that while weather parameters can enhance model performance, their impact varies across different models and should be carefully evaluated (Sillmann et al., 2017; Wu and Xue, 2024).

**Table 4.3**

*Impact of Feature Combinations on PM10 Prediction Model Performance*.

| Model | Feature Combinations | Train $R^2$ Score | Train RMSE | Train MAE | Validation $R^2$ Score | Validation RMSE | Validation MAE |
|---|---|---|---|---|---|---|---|
| ANN | Euclidean_D, Orientation | 0.9792 | 2.875 | 1.172 | 0.9775 | 2.231 | 1.1120 |
| | Wind speed, Temperature, Euclidean_D, Orientation | 0.9798 | 2.834 | 1.226 | 0.9793 | 2.143 | 1.0037 |
| SVR | Euclidean_D, Orientation | 0.9814 | 2.8874 | 1.3492 | 0.9739 | 2.3492 | 1.1315 |
| | Humidity, Euclidean_D, Orientation | 0.9817 | 2.8646 | 1.3292 | 0.9751 | 2.2940 | 1.1140 |
| LSTM | Euclidean_D, Orientation | 0.9837 | 2.5388 | 1.2689 | 0.9018 | 4.6475 | 3.5176 |
| | Temperature, Humidity, Euclidean_D, Orientation | 0.9827 | 2.6108 | 1.4414 | 0.9602 | 2.9566 | 1.8514 |
| Random Forest | Euclidean_D, Orientation | 0.4989 | 2.7161 | 2.1039 | 0.4672 | 2.8025 | 2.1755 |
| | Temperature, Humidity Euclidean_D, Orientation | 0.5988 | 2.4305 | 1.8839 | 0.4569 | 2.8294 | 2.2008 |

*Source: Research data, 2024.*

### 4.2.4 Average percentage prediction accuracy analysis

The study evaluated the generalizability of the four machine learning models (ANN, SVR, LSTM, and Random Forest) on unseen data. This involved predicting PM concentrations at points of interest (PoIs) within a 160-meter radius of a central sensor node, using various feature combinations. An average prediction accuracy metric was calculated for each PoI using equation 4.1.

$$Average\ Accuracy = \frac{1}{n}\sum_{i=1}^{n}\left(100 - \left(\frac{\left|y_{actual,i} - y_{predicted,i}\right|}{y_{actual,i}}\right) \times 100\right) \quad (4.1)$$

Where:

- $n$ is the number of data points.
- $y_{actual,i}$ is the actual observed value of the target variable for the $i_{th}$ data point in the testing dataset.
- $y_{predicted,i}$ is the predicted value of the target variable for the $i_{th}$ data point, as estimated by the ANN model.
- $|y_{actual,i} - y_{predicted,i}|$ is the Absolute Error.
- $\frac{|y_{actual,i} - y_{predicted,i}|}{y_{actual,i}}$ is the Relative Error.

To assess model performance, equation 4.1 was employed to calculate the absolute error between predicted and actual PM concentrations, expressed as a relative error in relation to the actual value. A 75% average accuracy threshold was empirically determined to identify the optimal model, independent of air quality health standards.

Tables 4.4 to 4.7 present the average prediction accuracy for each Point of Interest (PoI), corroborating the findings from Tables 4.1 to 4.3. Building upon the earlier observation of potential overfitting in LSTM and Random Forest models based on training and validation disparities, this analysis evaluates model performance on unseen data to assess their generalizability. The results indicate that the ANN model consistently outperformed SVR, LSTM, and Random Forest in predicting PM1.0, PM2.5, and PM10 concentrations within a 160-meter radius across all feature combinations. Conversely, the other models demonstrated substantial performance variations contingent on the PM scale and feature selection. This highlights the ANN model's superior ability to generalize and its robustness in predicting PM concentrations compared to the alternative models.

**Table 4.4**

*ANN Model Average Prediction Accuracy for PM1.0, PM2.5, and PM10.*

| FEATURES | PM1.0 | PM2.5 | PM10 |
|---|---|---|---|
| Euclidean_D, Orientation | 80.87 | 84.69 | 88.07 |
| Wind speed, Euclidean_D, Orientation | 84.05 | 86.91 | 85.74 |
| Temperature, Euclidean_D, Orientation | 77.29 | 88.5 | 81.77 |
| Humidity, Euclidean_D, Orientation | 85.33 | 88.73 | 82.96 |
| Temperature, Humidity, Euclidean_D, Orientation | 77.6 | 85.46 | 87.22 |
| Wind speed, Temperature, Euclidean_D, Orientation | 87.35 | 78.11 | 88.44 |
| Wind speed, Humidity, Euclidean_D, Orientation | 76.32 | 82.36 | 86.88 |
| Wind speed, Temperature, Humidity, Euclidean_D, Orientation | 76.01 | 74.34 | 81.75 |

*Source: Research data, 2024.*

Table 4.4 presents the average prediction accuracy of the ANN model for PM1.0, PM2.5, and PM10 concentrations across various feature combinations within a 160-meter radius from the central sensor node. The performance of the ANN model varies significantly across different feature combinations, reflecting the model's sensitivity to the inclusion of specific weather parameters.

As observed in Table 4.4, for PM1.0, the highest accuracy of 87.35% is achieved with the combination of wind speed, temperature, Euclidean_D, and Orientation, underscoring the critical role of these weather parameters in enhancing the ANN model's predictive capability. Similarly, for PM2.5, the combination of humidity and Euclidean_D alongside Orientation yields the highest accuracy of 88.73%, indicating the substantial impact of humidity on the model's performance. For PM10, the combination of wind speed, temperature, Euclidean_D, and Orientation again stands out, achieving an accuracy of 88.44%, highlighting the importance of both wind speed and temperature in predicting larger particulate matter concentrations.

These results demonstrate that the ANN model consistently outperforms other models, such as SVR, LSTM, and RandomForest, by surpassing the 75% set average prediction accuracy across all feature combinations. This superior performance is evident from the high accuracies achieved in predicting PM1.0, PM2.5, and PM10, with the ANN model effectively leveraging the combination of spatial and weather features to enhance its predictive accuracy.

Comparatively, the findings align with current studies that emphasize the influence of weather parameters on the performance of machine learning models for air quality prediction. Studies by (Li et al., 2017; Lin et al., 2023; Sun et al., 2019) highlight the critical role of dynamic weather conditions such as wind speed, temperature, and humidity in accurately predicting particulate matter concentrations. However, the specific feature combinations and their impact on model performance, as observed in this analysis, provide new insights that are less frequently addressed in existing literature. The absence of studies evaluating machine learning model performance across various feature combinations with changes in proximity from a reference point further underscores the novelty of these findings.

The ANN model's superior performance, as indicated by the results in Table 4.4, also highlights the model's robustness in handling complex, non-linear prediction tasks, which often surpasses the capabilities of SVR, LSTM, and RandomForest models. This observation is consistent with research by (Plocoste and Laventure, 2023; Sharma et al., 2021; Zickus et al., 2002), which suggest that neural networks tend to outperform support vector regression and other models in intricate prediction scenarios. Therefore, the results from Table 4.4 reinforce the ANN model's status as the best-performing model for predicting particulate matter concentrations, effectively utilizing a combination of spatial and weather features to achieve high prediction accuracies.

Table 4.5 presents the average prediction accuracy of the SVR model for PM1.0, PM2.5, and PM10 concentrations across various feature combinations within a 160-meter radius from the central sensor node. The SVR model's performance varies notably across different combinations, with accuracy levels showing the influence of including weather parameters as features. For PM1.0, the highest accuracy of 75.98% is achieved with the combination of Wind speed, Temperature, Euclidean_D, and Orientation. This suggests that wind speed and temperature are crucial for improving the SVR model's prediction accuracy for PM1.0. For PM2.5 and PM10, the inclusion of humidity alongside temperature and Euclidean_D proves to be the most impactful, with prediction accuracies of 78.31% and 80.46%, respectively. These results indicate that humidity is a significant factor in predicting larger particulate matter concentrations.

**Table 4.5**

*SVR Model Average Prediction Accuracy for PM1.0, PM2.5, and PM10.*

| FEATURES | PM1.0 | PM2.5 | PM10 |
|---|---|---|---|
| Euclidean_D, Orientation | 63.49 | 49.78 | 45.21 |
| Wind speed, Euclidean_D, Orientation | 49.59 | 54.14 | 47.4 |
| Temperature, Euclidean_D, Orientation | 75.82 | 78.21 | 80.37 |
| Humidity, Euclidean_D, Orientation | 75.46 | 78.31 | 80.46 |
| Temperature, Humidity, Euclidean_D, Orientation | 74.93 | 77.81 | 80.34 |
| Wind speed, Temperature, Euclidean_D, Orientation | 75.98 | 78.04 | 80.18 |
| Wind speed, Humidity, Euclidean_D, Orientation | 75.28 | 78.00 | 80.28 |
| Wind speed, Temperature, Humidity, Euclidean_D, Orientation | 74.89 | 77.69 | 80.14 |

*Source: Research data, 2024.*

As observer in Table 4.5, the impact of including weather parameters as training features is evident in the enhanced prediction accuracies across the different PM scales. Specifically, the inclusion of temperature and humidity consistently boosts the model's

performance for PM2.5 and PM10, highlighting their importance in capturing the variability of particulate matter concentrations. For PM1.0, wind speed and temperature stand out as the most influential parameters, corroborating the critical role of dynamic atmospheric conditions in predicting finer particulate matter. The consistent improvement in prediction accuracy with the addition of weather parameters underscores their value in the training process, contributing to more accurate and reliable model outputs.

These results complement the findings in Table 4.1, Table 4.2 and Table 4.3, where the SVR model's generalization on unseen data is shown to be robust yet slightly lower to that of the ANN model. While the SVR model performs significantly well across various feature configurations, with accuracies often around 75-80%, the ANN model demonstrates consistently higher accuracies above 80% for most feature combinations. This indicates that the ANN model has a superior ability to generalize and predict particulate matter concentrations across different conditions, reaffirming its status as the best-performing model overall.

This analysis aligns with other current studies that emphasize the importance of including weather parameters in machine learning models for air quality prediction. The findings support the notion that dynamic weather conditions, such as wind speed and temperature, are pivotal in accurately predicting PM concentrations, as highlighted by studies like those of (Aldegunde et al., 2022, 2022; Bekkar et al., 2021). However, the slightly lower generalization capability of the SVR model compared to the ANN model also resonates with research indicating that neural networks often outperform support vector regression in complex, non-linear prediction tasks by (Plocoste and Laventure, 2023; Sharma et al., 2021; Zickus et al., 2002). Overall, this analysis highlights the value

of comprehensive feature selection and the superior predictive power of the ANN model in the context of air quality monitoring.

**Table 4.6**

*LSTM Model Average Prediction Accuracy for PM1.0, PM2.5, and PM10.*

| FEATURES | PM1.0 | PM2.5 | PM10 |
|---|---|---|---|
| Euclidean_D, Orientation | 44.1 | 43.72 | 47.73 |
| Wind speed, Euclidean_D, Orientation | 52.55 | 58.16 | 62.15 |
| Temperature, Euclidean_D, Orientation | 74.53 | 66.59 | 80.74 |
| Humidity, Euclidean_D, Orientation | 74.16 | 74.13 | 72.95 |
| Temperature, Humidity, Euclidean_D, Orientation | 68.92 | 79.62 | 81.23 |
| Wind speed, Temperature, Euclidean_D, Orientation | 74.07 | 72.23 | 78.24 |
| Wind speed, Humidity, Euclidean_D, Orientation | 70.95 | 58.47 | 71.99 |
| Wind speed, Temperature, Humidity, Euclidean_D, Orientation | 70.21 | 71.37 | 74.06 |

*Source: Research data, 2024.*

The results in Table 4.6 present the average prediction accuracy of the LSTM model for PM1.0, PM2.5, and PM10 concentrations within a 160-meter radius from the central sensor node, revealing the model's performance across various feature combinations. The LSTM model's highest accuracy for PM1.0 is 74.53%, achieved with the combination of Temperature, Euclidean_D, and Orientation. This indicates that temperature is a critical feature influencing PM1.0 prediction accuracy. For PM2.5, the best performance is observed with Temperature, Humidity, Euclidean_D, and Orientation, resulting in an accuracy of 79.62%. Similarly, for PM10, the highest accuracy of 81.23% is also achieved with the inclusion of Temperature, Humidity, Euclidean_D, and Orientation. These findings underscore the importance of temperature

and humidity in enhancing the LSTM model's predictive capabilities for PM2.5 and PM10 concentrations.

Despite the LSTM model performing well with specific feature combinations, it generally exhibits lower prediction accuracy compared to the ANN and SVR models. This is evident from the overall average prediction accuracies, where the LSTM's best accuracies (74.53% for PM1.0, 79.62% for PM2.5, and 81.23% for PM10) are relatively lower than those achieved by ANN and SVR models across most feature sets. For instance, the ANN model consistently achieves prediction accuracies above 80% for all PM scales, and the SVR model reaches up to 75.98% accuracy, indicating better generalization and robustness in their predictions. These observations align with the results from Table 4.4.1, Table 4.4.2 and Table 4.4.3, where the ANN and SVR models showed superior performance metrics ($R^2$, RMSE, and MAE) compared to the LSTM, which struggled with generalization on unseen data.

The inclusion of weather parameters significantly impacts the LSTM model's prediction accuracy. Temperature consistently proves to be the most impactful feature for PM1.0, while the combination of Temperature and Humidity is crucial for predicting PM2.5 and PM10 concentrations. These findings highlight the sensitivity of the LSTM model to specific weather parameters, which enhance its ability to capture the temporal and spatial variability of particulate matter concentrations. However, even with these improvements, the LSTM model's accuracy remains lower than the ANN and SVR models, suggesting that while weather parameters are beneficial, the LSTM architecture may not be as effective in leveraging these features as the other models.

These results complement previous analyses by confirming the LSTM model's limitations in generalizing across various feature configurations. Despite achieving reasonable accuracies with optimal feature combinations, its overall performance is less

consistent compared to ANN and SVR models. This analysis aligns with other studies indicating that while LSTM models can capture temporal dependencies effectively, their performance may lag behind simpler, yet more robust models like ANN and SVR in certain applications (Plocoste and Laventure, 2023; Sharma et al., 2021; Zickus et al., 2002). Moreover, the relatively lower prediction accuracies of LSTM highlight the need for careful feature selection and model tuning to achieve optimal results in predicting air quality metrics.

While the LSTM model demonstrates significant improvements with specific weather parameters, its overall generalization is less effective compared to ANN and SVR models. The results underscore the importance of temperature and humidity in enhancing predictive accuracy but also highlight the inherent limitations of the LSTM model in spatial air quality prediction tasks. This analysis is consistent with current research, suggesting that while advanced models like LSTM have potential, traditional models may still offer superior performance in certain contexts (Han et al., 2021; Xiao et al., 2020).

The results in Table 4.7 also present the average prediction accuracy of the RandomForest model across various feature combinations within a 160-meter radius from the central sensor node for PM1.0, PM2.5, and PM10 scales. Across all feature configurations, the RandomForest model yielded prediction accuracies consistently below 65%, indicating its limited capability in accurately predicting PM concentrations spatially.

Despite varying feature combinations, the RandomForest model's performance remained notably lower compared to other models such as ANN, SVR, and LSTM, which achieved prediction accuracies predominantly above 80%, 74%, and 65%, respectively. This suggests that while RandomForest may offer certain advantages, such as simplicity

and scalability, its generalization on unseen data is comparatively lower, as evidenced by the consistently lower prediction accuracies.

Among the feature combinations, it appears that temperature and humidity plays a significant role in influencing the model's prediction accuracy for PM1.0, PM2.5, and PM10. Combining temperature and humidity with Euclidean_D and Orientation consistently resulted in prediction accuracies around 63% for PM1.0, 64% for PM2.5, and 62% for PM10. This underscores the importance of temperature as a crucial feature in training the model for accurate PM concentration prediction.

**Table 4.7**

*RandomForest Model Average Prediction Accuracy for PM1.0, PM2.5, and PM10.*

| **FEATURES** | **PM1.0** | **PM2.5** | **PM10** |
|---|---|---|---|
| Euclidean_D, Orientation | 64.48 | 64.16 | 62.23 |
| Wind speed, Euclidean_D, Orientation | 63.82 | 64.12 | 62.27 |
| Temperature, Euclidean_D, Orientation | 63.9 | 64.19 | 62.32 |
| Humidity, Euclidean_D, Orientation | 64.21 | 64.12 | 62.25 |
| Temperature, Humidity, Euclidean_D, Orientation | 63.94 | 64.16 | 62.33 |
| Wind speed, Temperature, Euclidean_D, Orientation | 63.53 | 64.19 | 62.38 |
| Wind speed, Humidity, Euclidean_D, Orientation | 63.76 | 64.10 | 62.29 |
| Wind speed, Temperature, Humidity, Euclidean_D, Orientation | 63.53 | 64.16 | 62.40 |

*Source: Research data, 2024.*

Notably, the inclusion of weather parameters, such as wind speed and humidity, alongside geometric features, did not notably enhance the RandomForest model's performance. Across different feature sets, the prediction accuracies remained relatively consistent, indicating that the addition of weather parameters did not significantly contribute to improving the model's predictive capability. This aligns with previous studies where RandomForest's performance was found to be relatively insensitive to the inclusion of environmental variables in certain contexts (Hart et al., 2020).

Comparatively, the RandomForest model's performance in this analysis differs from some current studies where it has shown promising results, particularly in scenarios with high-dimensional and noisy datasets. However, the lower prediction accuracies observed here could be attributed to the specific characteristics of the dataset and the complexity of predicting PM concentrations spatially, which may require more robust modeling approaches.

While the RandomForest model provides a robust framework for predictive modeling, its performance in spatial prediction of PM concentrations across various feature configurations within a 160-meter radius appears to be limited. Temperature emerges as the most impactful feature combination for training the model, indicating its crucial role in improving prediction accuracy. These results complement the findings from Table 4.1, Table 4.2 and Table 4.3, highlighting the importance of selecting appropriate models and features for accurate spatial prediction of PM concentrations.

### 4.2.5 Feature importance

Tables 4.8 to 4.10 delve into feature importance by ranking the average prediction accuracy of the four machine learning models (ANN, SVR, LSTM, and Random Forest) at points of interest (PoIs). This ranking is done from highest to lowest accuracy to understand how different feature combinations affect the models' ability to predict PM concentrations as the distance from the central sensor node changes. This analysis aims to identify the most significant weather parameters influencing PM distribution within a specific geographical area. Table 4.8 shows the training feature importance for various machine learning (ML) models in predicting PM1.0 concentrations across a 160-meter radius. The results for each model are discussed below.

**Table 4.8**

*Feature Importance Ranking for PM1.0 Prediction Models*

| Feature Combination | ANN Rank | SVR Rank | LSTM Rank | RandomForest Rank |
|---|---|---|---|---|
| Wind speed, Temperature, Euclidean_D, Orientation | 1 | 1 | 3 | 7 |
| Temperature, Euclidean_D, Orientation | 6 | 2 | 1 | 3 |
| Humidity, Euclidean_D, Orientation | 2 | 4 | 2 | 1 |
| Wind speed, Euclidean_D, Orientation | 3 | 3 | 4 | 5 |
| Temperature, Humidity, Euclidean_D, Orientation | 5 | 7 | 8 | 2 |
| Euclidean_D, Orientation | 4 | 8 | 7 | 4 |
| Wind speed, Humidity, Euclidean_D, Orientation | 7 | 5 | 6 | 6 |
| Wind speed, Temperature, Humidity, Euclidean_D, Orientation | 8 | 6 | 5 | 8 |

*Source: Research data, 2024.*

For the ANN model, the feature combination of wind speed, temperature, Euclidean_D, and Orientation achieved the highest average prediction accuracy of 87.35% (Table 4.4), highlighting the critical role of weather parameters. This aligns with existing research (Peralta et al., 2022; Xu and Zhang, 2020) that wind speed influences the dispersion and transport of particulate matter, while temperature affects particle composition and volatility. Including both these parameters significantly improved the model's accuracy.

While wind speed and temperature were the most impactful features, the ANN model also achieved reasonably high accuracy 80.87% (Table 4.4), with just Euclidean_D and Orientation. This suggests that even without weather data, the model can learn from spatial relationships. Lower-ranked combinations including humidity with other parameters showed reduced accuracy, indicating that while these parameters are

significant individually, their combined effect is less influential than wind speed and temperature together.

Similar to the ANN model, the SVR model achieved the highest prediction accuracy (75.98%, Table 4.5) with the combination of wind speed, temperature, Euclidean_D, and Orientation. This again highlights the importance of wind speed and temperature in capturing PM dynamics, as supported by studies (Li et al., 2017) which explain how wind speed influences PM transport and dilution, while temperature affects chemical reactions and formation processes. Interestingly, adding multiple weather parameters together did not significantly improve performance beyond a certain point, suggesting a saturation effect. This aligns with research (Badura et al., 2018) that emphasizes careful consideration of feature selection to avoid overfitting and ensure robust predictions.

The LSTM model exhibited the most significant variation in performance based on feature combinations. The highest accuracy (74.53%, Table 4.6) was achieved with temperature, Euclidean_D, and Orientation, suggesting that temperature plays a crucial role in capturing the temporal dependencies inherent in PM1.0 data. This might be because PM1.0 particles are more sensitive to temperature variations. Including humidity also resulted in high accuracy (74.16%, Table 4.6), indicating its importance as well.

While wind speed and Euclidean_D alone achieved a lower accuracy (70.95%, Table 4.6) compared to the combination with temperature, including all three weather parameters did not significantly boost performance (70.21%, Table 4.6). This suggests a diminishing return on adding weather features beyond a certain point, similar to the SVR model. The importance of temperature in LSTM training aligns with current research (Gong et al., 2015) on how temperature and humidity influence PM concentrations through atmospheric effects. However, unlike the other models, the LSTM model's validation results raise concerns about generalization ability. Although training and validation

metrics were relatively close, higher errors on validation data suggest the model may struggle with unseen data, potentially due to overfitting.

The Random Forest model achieved the highest prediction accuracy (63.94%, Table 4.7) with the combination of temperature, humidity, Euclidean_D, and Orientation. This indicates that although slightly lower than when the model was trained without the inclusion of weather parameters (64.48%, Table 4.7), temperature and humidity is the most crucial feature for this model. Similar to the other models, including other weather parameters like humidity and wind speed offered some improvement, but to a lesser extent. This aligns with research (Li et al., 2017; Chen et al., 2019) that temperature variations directly influence particle dispersion and atmospheric reactions. The model's performance on unseen data suggests limitations in generalization. While training and validation metrics were closer than for other models, higher errors on validation data indicate it may not perform well with new data. This is consistent with findings (Cofre-Martel et al., 2021) that Random Forest models, while effective for complex datasets, can struggle with overfitting and fail to capture the underlying data distribution on new datasets.

These findings demonstrate the complex interplay between feature importance and model performance in PM concentration prediction. While temperature emerged as a significant factor for all models, their performance varied based on feature combinations and generalization ability. Careful feature selection, model evaluation, and consideration of potential overfitting are crucial for robust air quality monitoring applications

**Table 4.9**

*Feature Importance Ranking for PM2.5 Prediction Models*.

| Feature Combination | ANN Rank | SVR Rank | LSTM Rank | RandomForest Rank |
|---|---|---|---|---|
| Humidity, Euclidean_D, Orientation | 1 | 1 | 2 | 5 |
| Wind speed, Euclidean_D, Orientation | 3 | 3 | 3 | 4 |
| Wind speed, Humidity, Euclidean_D, Orientation | 6 | 2 | 5 | 7 |
| Euclidean_D, Orientation | 5 | 5 | 1 | 1 |
| Temperature, Euclidean_D, Orientation | 2 | 7 | 7 | 3 |
| Temperature, Humidity, Euclidean_D, Orientation | 4 | 4 | 6 | 2 |
| Wind speed, Temperature, Euclidean_D, Orientation | 7 | 8 | 8 | 6 |
| Wind speed, Temperature, Humidity, Euclidean_D, Orientation | 8 | 6 | 4 | 8 |

*Source: Research data, 2024.*

Table 4.9 summarizes the training feature importance for various machine learning models in predicting PM2.5 concentrations across a 160-meter radius. The Artificial Neural Network (ANN) model achieved the highest average prediction accuracy 88.73% (Table 4.4) with the feature combination of humidity, Euclidean_D, and Orientation. This aligns with existing research emphasizing the significant influence of humidity on PM2.5 prediction in ANN models (Zang et al., 2019). Humidity significantly impacts particulate matter dispersion and concentration through processes like hygroscopic growth.

Following closely behind the ANN model was the Support Vector Regression (SVR) model, with a top accuracy of 78.31% (Table 4.5) for the same feature combination

(humidity, Euclidean_D, and Orientation). Similar to the ANN model, the SVR model highlights the importance of humidity in capturing PM2.5 concentrations.

The Long Short-Term Memory (LSTM) model exhibited a different pattern. Here, the most influential combination (79.62% accuracy, Table 4.6) included only spatial features (Euclidean_D and Orientation). This suggests the LSTM model's strength in learning from the spatial distribution of pollutant sources. However, combinations involving humidity and wind speed alongside spatial features also demonstrated notable predictive power, which is consistent with prior research on the crucial role of these meteorological factors in PM2.5 dynamics (Won et al., 2021; S. Zhang et al., 2023).

The Random Forest model achieved a lower overall accuracy (64.16%, when trained with inclusion, and without the inclusion of the either parameters as training features, Table 4.7) compared to the other models. The most effective combination included temperature, humidity, Euclidean_D, and Orientation. This finding aligns with current studies highlighting the importance of wind speed and humidity in PM prediction models, as they significantly influence particulate matter dispersion and behavior (Won et al., 2021).

Despite some variations in feature importance across the models, all models highlighted the significant influence of weather parameters (humidity, temperature, and wind speed) on PM2.5 prediction. This aligns with existing research demonstrating the crucial role of these factors in PM2.5 concentrations across different spatial scales.

Table 4.10 presents the feature importance analysis for ANN, SVR, LSTM, and Random Forest models in predicting PM10 concentrations across a 160-meter radius. The analysis highlights the varying impacts of different feature combinations on model accuracy.

The ANN model demonstrated strong predictive capability for PM10 concentrations (average accuracy: 88.44%, Table 4.4), with weather parameters (wind speed,

temperature) and spatial features (Euclidean_D, Orientation) playing essential roles. The most influential combination included wind speed (essential for dispersion), temperature, Euclidean_D, and Orientation (capturing spatial variations), achieving an accuracy of 88.44%. This aligns with existing research emphasizing the importance of both weather and spatial features in air quality modeling (Shiferaw et al., 2023).

**Table 4.10**

*Feature Importance Ranking for PM10 Prediction Models*.

| Feature Combination | ANN Rank | SVR Rank | LSTM Rank | RandomForest Rank |
|---|---|---|---|---|
| Humidity, Euclidean_D, Orientation | 6 | 2 | 2 | 3 |
| Temperature, Euclidean_D, Orientation | 7 | 3 | 1 | 3 |
| Wind speed, Humidity, Euclidean_D, Orientation | 4 | 1 | 5 | 5.5 |
| Temperature, Humidity, Euclidean_D, Orientation | 3 | 4 | 6 | 1 |
| Wind speed, Temperature, Euclidean_D, Orientation | 1 | 5 | 3 | 7.5 |
| Wind speed, Temperature, Humidity, Euclidean_D, Orientation | 8 | 6 | 4 | 6 |
| Euclidean_D, Orientation | 2 | 8 | 8 | 5.5 |
| Wind speed, Euclidean_D, Orientation | 5 | 7 | 7 | 7.5 |

*Source: Research data, 2024.*

The SVR model's performance (average accuracy: 80.46%, Table 4.5) also indicated the influence of different feature combinations. The most impactful combination included wind speed, humidity, Euclidean_D, and Orientation, achieving an accuracy of 80.46%. This aligns with current studies highlighting the importance of wind speed for transport and humidity for particle dynamics (Saiohai et al., 2023). Interestingly, wind speed showed a lower impact when combined solely with Euclidean_D and Orientation (47.4%

and 45.21% accuracy, respectively, Table 4.5), suggesting its effectiveness increases when combined with other relevant features like humidity.

The LSTM model's performance (average accuracy: 81.23%, Table 4.6) emphasized the role of spatial features and temperature. The most influential combination included temperature, Euclidean_D, and Orientation (accuracy: 81.23%). Temperature likely influences chemical reactions and particle behavior (Batmunkh et al., 2013). Humidity also played a significant role, with the combination of humidity, Euclidean_D, and Orientation achieving an accuracy of 80.74% (Table 4.6). However, the validation results raised concerns about generalization ability. While training and validation metrics were relatively close, higher RMSE and MAE values during validation suggest potential overfitting.

The Random Forest model's performance (Table 4.7) presented lower average accuracy results. The most influential combination included all features (temperature, humidity, Euclidean_D, and Orientation), achieving an accuracy of 62.33%. This contradicts the typical preference of Random Forest models for simpler feature sets. However, it suggests that for PM10 prediction, comprehensive consideration of weather and spatial features might be beneficial. Interestingly, simpler combinations with only spatial features showed lower accuracies (around 62%, Table 4.7), highlighting the importance of weather parameters. Despite relatively close training and validation metrics, higher validation RMSE and MAE values suggest potential overfitting issues (Hawkins, 2004; Kolluri et al., 2020).

Through the analysis, all models highlighted the importance of weather parameters (wind speed, temperature, humidity) and spatial features (Euclidean_D, Orientation) in PM10 concentration prediction. However, the ranking of feature importance and overall accuracy varied across models. ANN and SVR achieved higher accuracies than LSTM

and Random Forest. Generalization ability was a concern for LSTM and Random Forest models, requiring further tuning and validation for real-world application. These findings align with current research emphasizing the complex interplay between weather conditions, spatial factors, and PM10 concentrations.

### 4.2.6 Validating the ANNS model as the best performer

The pre-deployment analysis reveals the performance of various machine learning models in predicting PM concentrations at the PoI, with a specific focus on the ANN, SVR, LSTM, and RandomForest models. The analysis highlights the ANN model's superior performance in predicting the spatial distribution of PM1.0, PM2.5, and PM10 concentrations across different feature combinations within a 160-meter radius from the central sensor node.

For PM1.0, the ANN model trained with the combination of wind speed, temperature, Euclidean_D, and Orientation emerged as the best-performing model. It achieved the highest validation $R^2$ score of 0.9756 and the lowest RMSE (1.7201) and MAE (0.7952) values, indicating its effectiveness in spatially predicting PM concentrations. The inclusion of wind speed and temperature significantly enhanced the model's performance, demonstrating the critical importance of these weather parameters in achieving accurate predictions. This combination underscores the ANN model's capability to capture the complex dynamics of PM1.0 distribution, surpassing other models like SVR, LSTM, and RandomForest in terms of predictive accuracy.

Among the models evaluated, the ANN model also proved to be the most suitable for predicting PM2.5 concentrations. It achieved the highest training and validation $R^2$ scores of 0.9805 and 0.9789, respectively, along with the lowest RMSE (2.2139) and MAE (1.0012). These metrics highlight the ANN model's robustness and precision in handling PM2.5 data, outperforming SVR, LSTM, and RandomForest models. The

ability of the ANN model to consistently achieve high $R^2$ scores while maintaining low error rates validates its selection as the best-performing model for PM2.5 prediction. The model's enhanced performance with the inclusion of weather parameters such as humidity and Euclidean_D further emphasizes the importance of incorporating dynamic atmospheric conditions in predictive modeling.

In predicting PM10 concentrations, the ANN model achieved a validation $R^2$ of 0.9775, indicating a strong fit. This was further improved to 0.9793 upon incorporating wind speed and temperature, accompanied by a decrease in RMSE from 2.2310 to 2.1430 and MAE from 1.1120 to 1.0037. These improvements highlight the model's enhanced precision and its ability to minimize prediction errors. The minimal deviation from actual PM10 concentrations, with an average prediction error of 1.0037 μg/m³, underscores the significance of wind speed and temperature in refining the ANN model's performance. This analysis confirms that the ANN model's predictive accuracy for PM10 is significantly enhanced by the inclusion of critical weather parameters, making it the most reliable model among those tested.

The instantaneous average prediction accuracy further validates the ANN model's superiority. For PM1.0, the highest accuracy of 87.35% is achieved with the combination of wind speed, temperature, Euclidean_D, and Orientation, underscoring the importance of these parameters in boosting the model's predictive capability. Similarly, for PM2.5, the combination of humidity and Euclidean_D alongside Orientation yields the highest accuracy of 88.73%, indicating the substantial impact of humidity on the model's performance. For PM10, the combination of wind speed, temperature, Euclidean_D, and Orientation stands out with an accuracy of 88.44%, highlighting the importance of both wind speed and temperature in predicting larger particulate matter concentrations.

This analysis establishes the ANN model as the best-performing and most suitable model for predicting PM concentrations across all feature combinations and PM scales. The superior $R^2$ scores, RMSE, and MAE metrics consistently position the ANN model ahead of SVR, LSTM, and RandomForest, validating its selection for further post-deployment analysis on unseen data. The focus of the post-deployment analysis will be on the selected ANN model, given its demonstrated impressive performance in predicting PM1.0, PM2.5, and PM10 concentrations across various feature combinations and its capacity to generalize effectively from the training data to unseen data.

### 4.3 Assessment of Models' Prediction Accuracies with Distance

The post-deployment evaluation centered on the selected ANN model's capability to predict PM concentrations (PM1.0, PM2.5, and PM10) within a 160-meter radius from a central sensor node using unseen data. To evaluate the influence of weather data on prediction accuracy, two model configurations were analyzed. The first model was trained solely with geometric features, including Euclidean_D and Orientation. The second model incorporated both geometric features and the most impactful weather parameters identified during the pre-deployment analysis: wind speed and temperature for PM1.0 and PM10, and humidity for PM2.5.

Three techniques were used to assess the model's performance. The first technique, Instantaneous Concentration Comparison, involved directly comparing actual and predicted PM concentrations at various points within the 160-meter radius, providing immediate insights into prediction accuracy. The second technique, Signal Energy Analysis, evaluated overall prediction accuracy by calculating the energy of both actual and predicted PM concentration profiles across the entire radius. The percentage error in signal energy, derived from equation 3.12, measured the difference between these profiles, highlighting how closely the predicted profile matched actual variations in PM

concentration over space. This approach is particularly useful for discrete data, where each point corresponds to a specific location within the radius.

The third technique, Moving Average Analysis, smoothed out short-term fluctuations in the instantaneous prediction accuracy data, using equation 4.1. This analysis revealed trends in accuracy across varying distances from the sensor node, offering a clearer picture of the model's performance as distance changes. The results of these analyses are depicted in Figures 4.1, 4.2, 4.3, 4.4, 4.5, and 4.6, providing insights into the ANN model's effectiveness and the impact of incorporating weather parameters on its predictive capabilities.

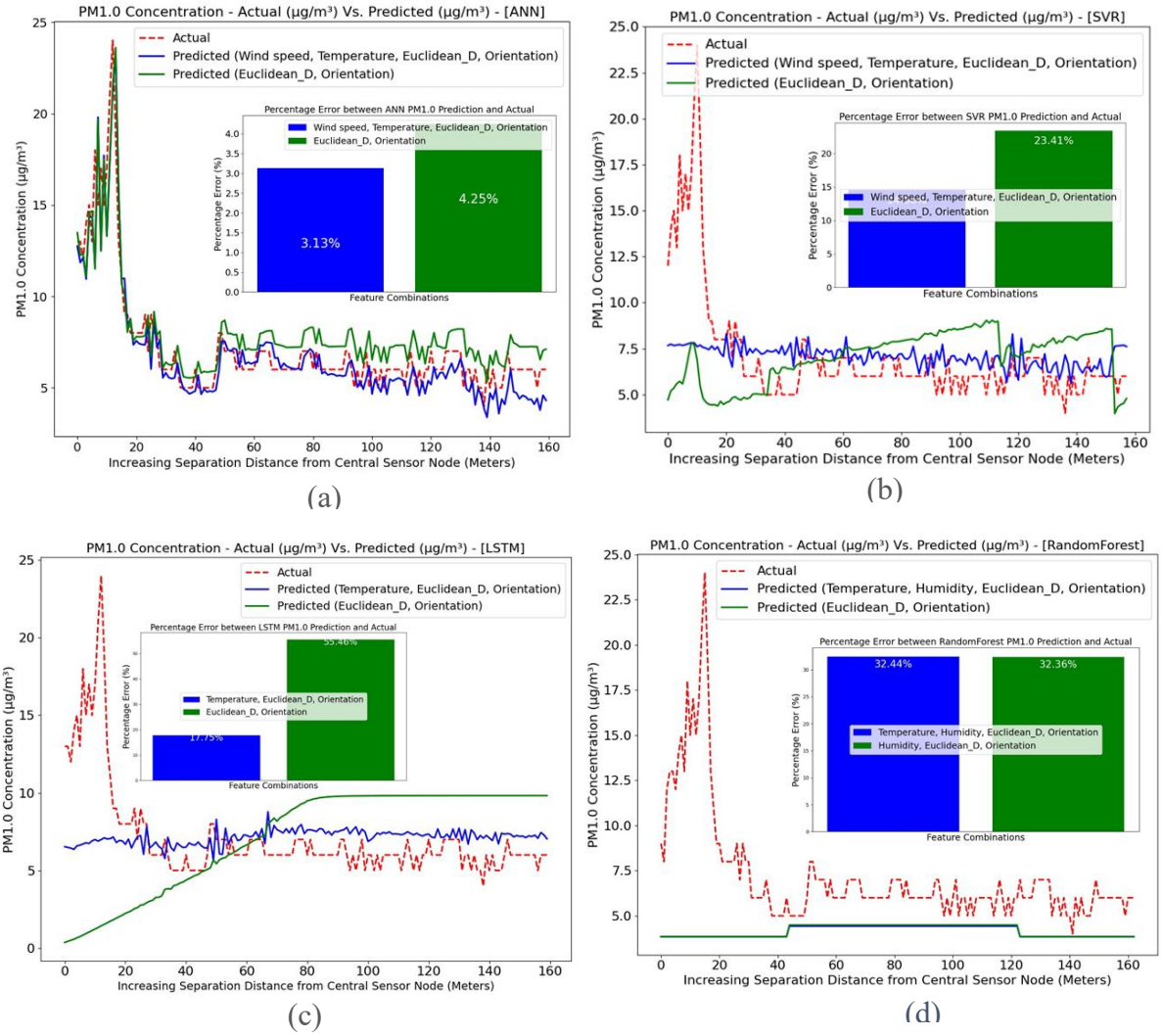
### 4.3.1 Post-deployment results

Figure 4.1 illustrates the alignment between predicted and actual PM1.0 concentrations within the 160-meter radius. The ANN model trained with both geometric features and weather parameters (wind speed and temperature) exhibits a closer alignment with actual PM1.0 concentrations compared to the model using only geometric features. This improved alignment suggests that incorporating weather data enhances the model's ability to capture the spatial distribution of PM1.0.

This finding is further supported by the signal energy analysis. The model with weather data achieved a lower signal energy error, reflected in a reduced percentage error of around 3.13% compared to 4.25% for the model without weather data. This signifies a smaller divergence between the predicted and actual PM1.0 concentration profiles across the entire radius.

**Figure 4.1**

*Prediction Accuracy of ANN Model for PM1.0 Concentrations across a160-Meter*

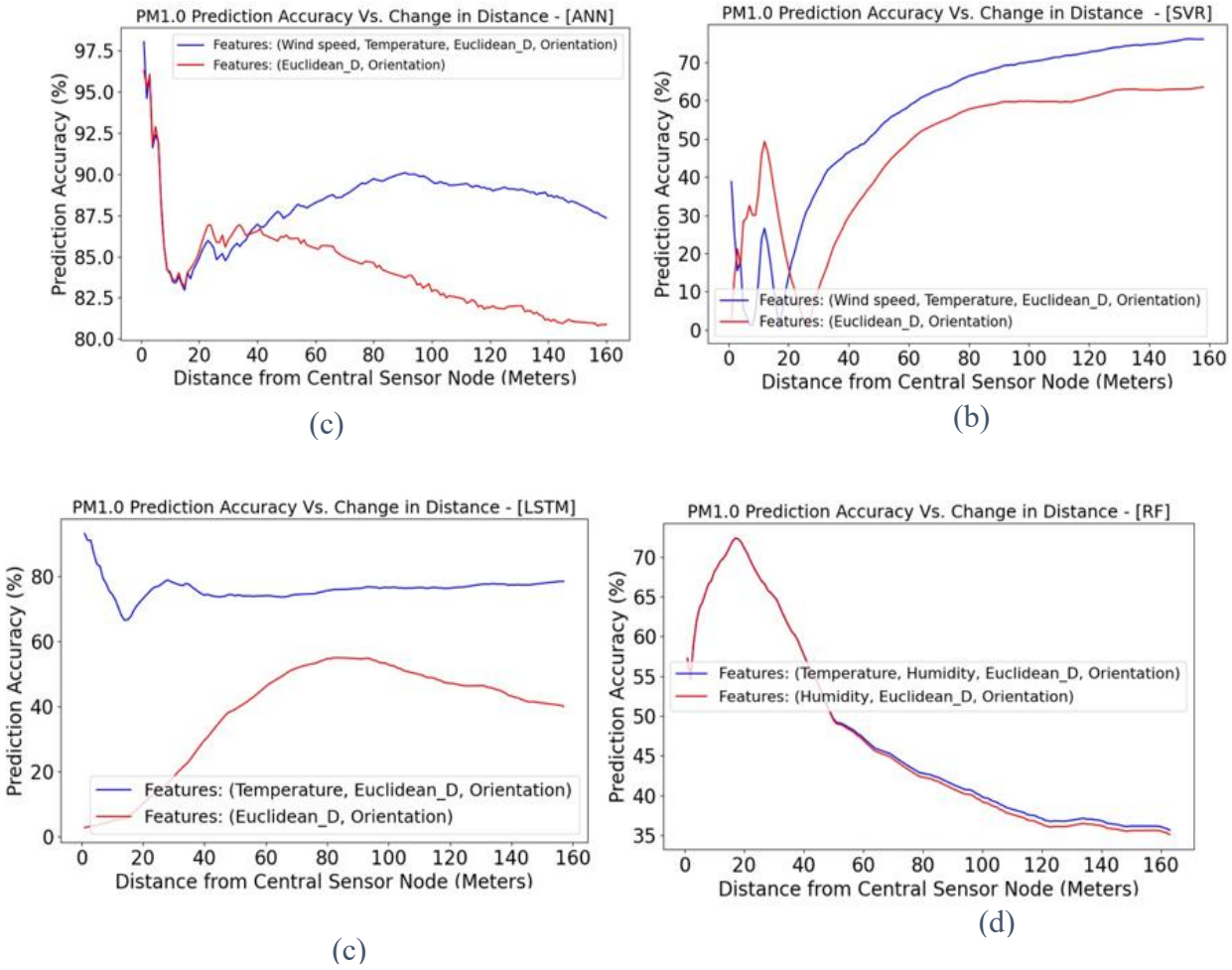



*Source: Research Data 2024.*

Figure 4.2 illustrates the smoothened prediction accuracy profile of the Artificial Neural Network (ANN) model for PM1.0 concentrations across increasing distances from the central sensor node).

**Figure 4.2**

*Smoothened PM1.0 ANN Prediction Accuracy with Change in Distance.*



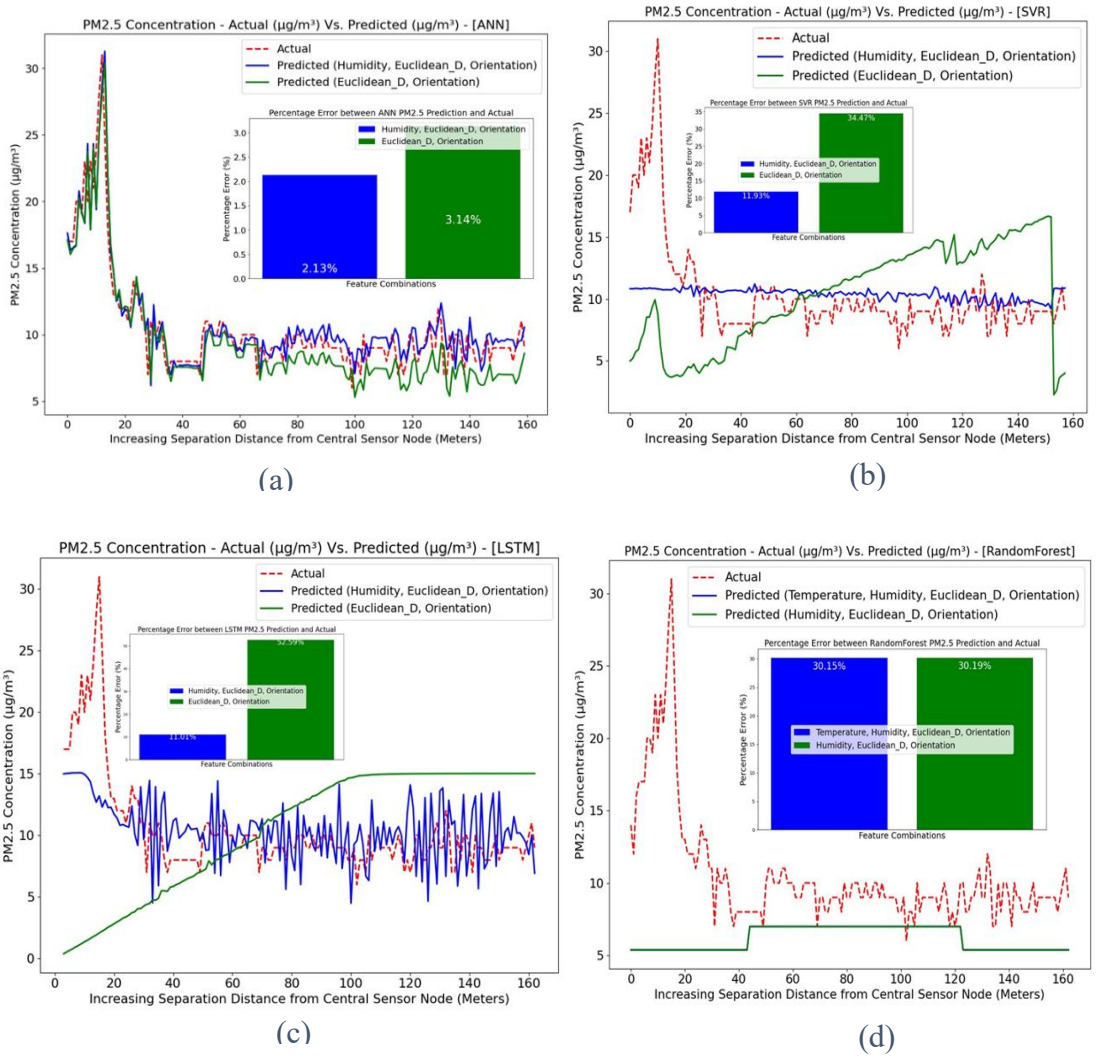

*Source: Research Data 2024.*

Figure 4.2 provides further support for the findings depicted in Figure 4.1. It depicts smoothed prediction accuracy profiles for both models across the radius. The model incorporating weather data (blue line) maintains a consistently higher and more stable accuracy level compared to the model without weather data (red line). Notably, the red line shows a dip in accuracy between 10 and 25 meters from the sensor node. This dip aligns with the potential challenges arising from uneven training data distribution across spatial areas. It underscores the importance of well-balanced datasets in achieving robust model performance across the entire prediction range.

Figure 4.3 illustrates the alignment between predicted and actual PM2.5 concentrations within a 160-meter radius. The ANN model trained with both geometric features and the impactful weather parameter (humidity) showed a better alignment with the actual PM2.5 concentrations compared to the model using only geometric features. This is evident from the closer correspondence between the predicted and actual concentration profiles.

**Figure 4.3**

*Prediction Accuracy of ANN Model for PM2.5 Concentrations across a 160-Meter Radius from Central Sensor Node.*

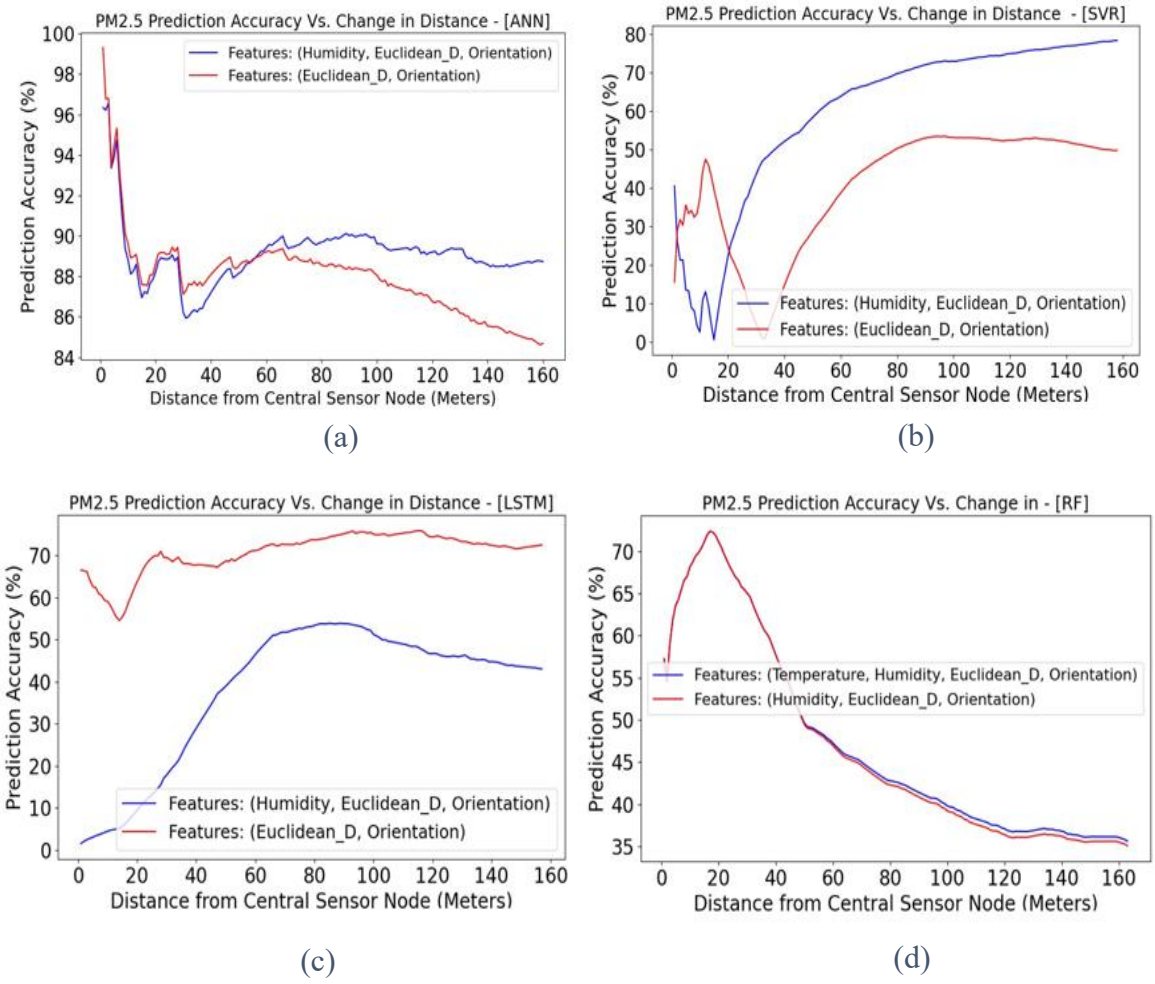



*Source: Research data, 2024.*

Additionally, the inclusion of weather data in the model led to a lower signal energy error, as demonstrated by a reduced percentage error (around 2.13% compared to 3.14%). This indicates that the model with weather data captured the overall variations in PM2.5 concentration across the spatial range more accurately.

**Figure 4.4**

*Smoothened PM2.5 ANN Prediction Accuracy with Change in Distance.*




*Source: Research data, 2024.*

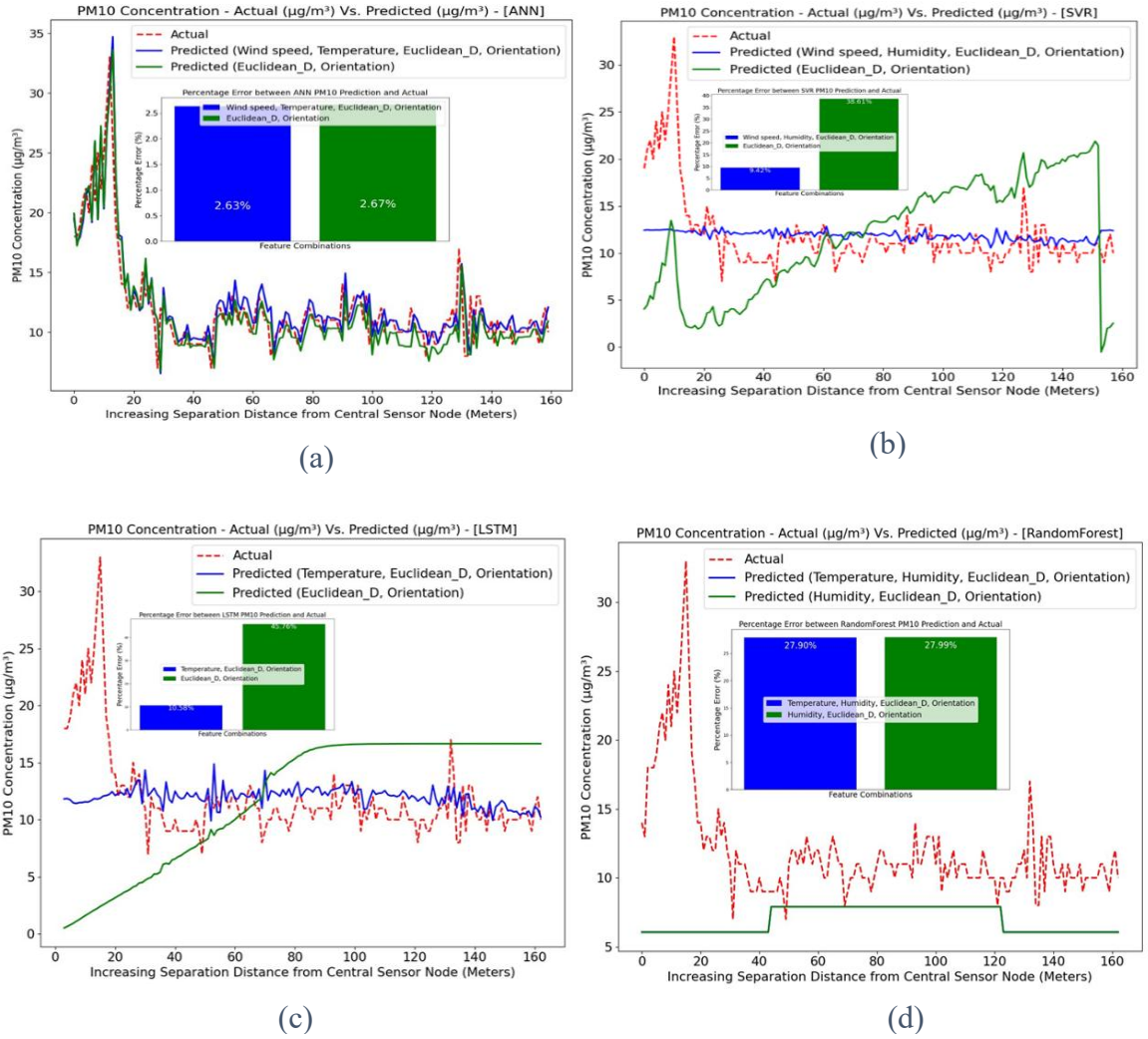
Figure 4.4 illustrates the smoothened prediction accuracy profile of the Artificial Neural Network (ANN) model for PM2.5 concentrations across increasing distances from the central sensor node).

Supporting Figure 4.3, Figure 4.4 depicts the smoothed prediction accuracy profiles for PM2.5 concentrations. The ANN model with weather data (blue line) maintained higher and more consistent accuracy across distances compared to the model without weather data (red line). The red line shows a dip in accuracy, particularly pronounced between the $20^{th}$ and $30^{th}$ meters from the sensor node. This dip suggests potential challenges arising from uneven training data distribution across spatial domains. It highlights the importance of a well-balanced dataset for achieving robust model performance across the entire area of interest.

Figure 4.5 illustrates the alignment between predicted and actual PM10 concentrations within the 160-meter radius. The ANN model trained with both geometric features and weather parameters (wind speed and temperature) demonstrates a closer alignment with the actual PM10 concentrations compared to the model using only geometric features. This improved alignment is reflected in a lower signal energy error. As shown in the figure, the percentage error is around 2.63% for the model with weather data, compared to 2.67% for the model without. This signifies that incorporating weather data helps the model capture the spatial variations in PM10 concentrations more accurately.

**Figure 4.5**

*Prediction Accuracy of ANN Model for PM10 Concentrations across a 160-Meter Radius from Central Sensor Node.*



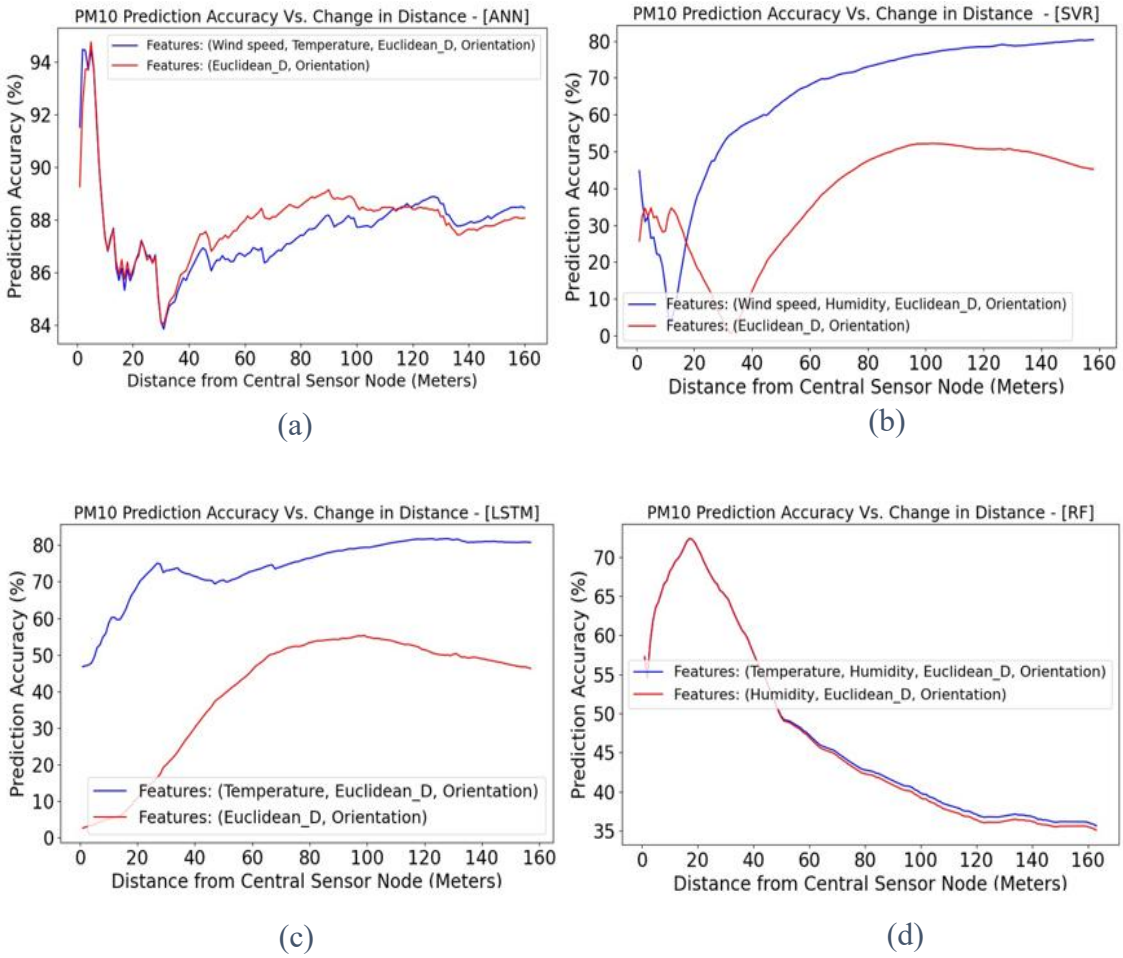

*Source: Research data, 2024.*

Supporting Figure 4.5, Figure 4.6 depicts the smoothed prediction accuracy profiles for both model configurations. While the accuracy profiles are generally close, the model with weather data (blue line) maintains a consistently higher accuracy across most distances compared to the model without weather data (red line). Notably, the red line shows a dip in accuracy around 30 meters from the central sensor node. This dip suggests potential challenges arising from uneven training data distribution across spatial

areas. A well-balanced dataset encompassing a wider range of spatial variations is crucial for ensuring robust model performance across the entire 160-meter radius.

**Figure 4.6**

*Smoothened PM10 ANN Prediction Accuracy with Change in Distance.*




*Source: Research data, 2024.*

The results in Figure 4.6 illustrates the smoothened prediction accuracy profile of the Artificial Neural Network (ANN) model for PM10 concentrations across increasing distances from the central sensor node.

### 4.3.2 Post-deployment results discussion

The post-deployment analysis of the ANN model performance highlights its robustness in predicting PM1.0, PM2.5, and PM10 concentrations within a 160-meter radius of a central sensor node. Figures 4.1, 4.3, and 4.5 depict the alignment between predicted and

actual concentrations for these particulate matter scales, showing that the ANN model trained with both geometric features and weather parameters (wind speed, temperature, and humidity) aligns more closely with actual measurements compared to the model using only geometric features. This is evidenced by lower signal energy errors, with percentage errors of around 3.13% for PM1.0, 2.13% for PM2.5, and 2.63% for PM10 when weather data is included, compared to 4.25%, 3.14%, and 2.67%, respectively, for models without weather data. These results underscore the enhanced ability of the ANN model to capture the spatial distribution of particulate matter concentrations accurately when weather parameters are integrated.

Figures 4.2, 4.4, and 4.6 further support these findings by providing smoothed prediction accuracy profiles. The ANN model incorporating weather data (blue line) maintains consistently higher and more stable accuracy levels across the radius compared to the model without weather data (red line). Notably, the red line shows dips in accuracy, such as between the $10^{th}$ and $25^{th}$ meters for PM1.0, the $20^{th}$ and $30^{th}$ meters for PM2.5, and around 30 meters for PM10. These dips indicate potential challenges due to uneven training data distribution across spatial domains, emphasizing the importance of well-balanced datasets for achieving robust model performance.

The comparison of signal energy percentage errors and the instant alignment of predictions with actual data reveal that including weather parameters significantly improves the ANN model's performance across all particulate matter scales. For instance, the lower and closely matched signal energy percentage errors for PM10 ANN model's performance at 2.63% (with weather data) and 2.67% (without weather data) correspond to the close average prediction accuracy reported in Table 4.7, validating the consistency of the model's performance with different feature configurations.

The use of geometric features and feature engineering, combined with meteorological parameters, significantly enhances the model's performance. The ANN model's instantaneous performance remains relatively high, above 86%, as depicted in Figures 4.1 for PM1.0, 7 for PM2.5, and 9 for PM10. This high level of accuracy across different distances from the central sensor node confirms the robustness and reliability of the ANN model for predicting particulate matter concentrations.

These findings align with current studies emphasizing the impact of meteorological factors on air quality predictions. Research by (Li et al., 2019; Tai et al., 2011), and others highlights the critical role of weather parameters such as wind speed and temperature in improving model performance. However, the unique approach of this study in transforming GPS spatial data to radians for distance computations and incorporating these transformed geometric features alongside weather data distinguishes it from other studies. Most existing research has not extensively explored distance-based feature engineering in the context of air quality prediction, making this study's methodology and findings particularly noteworthy.

The superior performance of the ANN model compared to LSTM and SVR can be attributed to its inherent ability to learn complex, nonlinear relationships, making it well-suited for modeling air pollution dynamics. The experimental data exhibits strong spatial dependencies influenced by meteorological factors, which ANN effectively captures by leveraging its multi-layered structure. Unlike LSTM, which excels in sequential time-series forecasting but may struggle with purely spatial relationships, ANN adapts more flexibly to structured spatial features without relying on temporal dependencies. Similarly, while SVR performs well in capturing simple relationships, it lacks the capacity to fully exploit the interactions between geometric features and weather parameters, leading to reduced accuracy in predicting PM concentrations across varied

spatial domains. The ANN model's ability to integrate and interpret both physical and spatial characteristics of the data accounts for its consistently higher accuracy and lower error rates, as demonstrated in Figures 4.1 through 4.6. This confirms its suitability for modeling particulate matter concentrations, particularly in heterogeneous environments where multiple influencing factors interact in a nonlinear manner.

# CHAPTER FIVE: CONCLUSION, RECOMMENDATIONS AND PUBLICATION

## 5.1 Conclusion

The primary goal of this research was to apply machine learning models for estimating particulate matter pollutants in the air and inferring their spatial distribution within a specified radius. Current PM monitoring systems are often limited to single-point readings and lack the capability to map PM distribution effectively. Deploying a dense network of sensors for multi-point data collection is resource-intensive and impractical. To address these challenges, this study utilized an IoT-enabled sensor system and leveraged machine learning models to predict PM concentrations over a 160-meter radius from a central sensor node.

The first objective involved utilizing an IoT-enabled sensor system to measure particulate pollutant concentrations, spatial data (Latitude and Longitude), and meteorological parameters (Wind speed, ambient Temperature, and Humidity). The sensor nodes collected raw PM concentration data, which was preprocessed to extract relevant input features (Wind speed, Temperature, Humidity, Euclidean_D, and Orientation) and target features (PM1.0, PM2.5, and PM10). These features were used to train and validate four machine learning models: ANN, SVR, LSTM, and RandomForest.

The second objective focused on assessing the impact of integrating meteorological parameters as input features for training machine learning models to predict the spatial distribution of particulate matter concentrations within a specified geographic area using limited local measurements. This assessment was conducted through pre-deployment analysis, where the ANN model demonstrated superior performance among the four evaluated models. The models were trained to predict PM values at a given distance from a reference point, referred to as the Point of Interest (PoI). The model inputs included

PM values at the reference point and geometric features such as Euclidean_D and Orientation. Additional models were trained using PM values at the reference point, geometric features, and meteorological data (wind speed, temperature, and humidity). Performance was evaluated using $R^2$ scores, RMSE, and MAE.

The third objective validated the ANN model's performance through post-ML model deployment analysis. The model's predictive accuracy was tested on unseen data to estimate PM concentrations within a 160-meter radius from the central sensor node. The evaluation compared ANN models trained with only geometric features to those incorporating both geometric and meteorological features. The inclusion of weather parameters significantly enhanced accuracy, as evidenced by lower signal energy percentage errors and improved instantaneous prediction precision across all PM scales.

Results indicated that the ANN model trained with a combination of geometric features and meteorological parameters achieved lower signal energy errors and higher prediction accuracy compared to the model trained with only geometric features. For example, when weather data was included, percentage errors were approximately 3.13% for PM1.0, 2.13% for PM2.5, and 2.63% for PM10, whereas models trained without weather data yielded errors of 4.25%, 3.14%, and 2.67%, respectively. Figures 4.1, 4.3, and 4.5 illustrate the improved alignment between predicted and actual PM concentrations, while Figures 4.2, 4.4, and 4.6 present smoothed prediction accuracy profiles, demonstrating consistently higher accuracy levels when weather data was included.

The novelty of this study lies in examining the impact of meteorological parameters on ML-based PM prediction, employing a feature engineering approach that transforms raw GPS data into radians for distance computations, and integrating geometric features (Euclidean_D and Orientation) in model training. This approach differs from existing studies, which often overlook distance-based feature engineering. The inclusion of wind

speed, temperature, and humidity significantly improved the ANN model's predictive performance, as indicated by high $R^2$ scores, low RMSE, and low MAE values.

Compared to existing research, the findings align with studies by Qiao et al. (2022), Tai et al. (2011), and Won et al. (2021), which highlight the influence of meteorological factors on air quality predictions. However, this study uniquely incorporates transformed geometric features alongside weather data, offering new insights into the advantages of combining these features for more accurate air quality modeling.

Overall, this research demonstrates the effectiveness of machine learning models, particularly ANN, in predicting PM concentrations and their spatial distribution within a specified radius. The findings underscore the importance of integrating meteorological parameters and geometric features to enhance model performance, providing a cost-effective solution for comprehensive air quality monitoring. This approach mitigates the limitations of single-point monitoring systems and generates valuable data for environmental regulators, policymakers, and future research initiatives, contributing to improved air quality management and public health protection.

### 5.2 Recommendations

Expanding the number of reference points within the study area presents an opportunity to further enhance the accuracy and reliability of machine learning models for PM concentration prediction. The inclusion of additional sensor nodes would increase spatial coverage, enabling models to capture finer variations in pollutant distribution across different microenvironments. A denser network of reference points would reduce interpolation gaps and improve model generalizability, particularly in regions where pollutant dispersion is influenced by localized meteorological and topographical factors. Future research should explore the impact of multiple reference nodes by employing similar approaches as applied in this study while investigating how the increased data

granularity refines model predictions. Comparative analyses between models trained on sparse and dense sensor networks would provide valuable insights into optimal deployment strategies for air quality monitoring.

Incorporating distance-based features, such as Euclidean distance transformed using the haversine function, enhances model accuracy by capturing the spatial distribution of particulate matter concentrations. These features provide essential insights into how pollutant levels vary across different locations, particularly in relation to emission sources. Since PM concentrations generally decline with distance from the source, integrating spatial data ensures that models better reflect real-world dispersion patterns, improving predictive accuracy and reliability.

Addressing uneven data distribution through distance-based data filling techniques is essential for ensuring unbiased model predictions across different spatial domains. When sensor coverage is sparse in certain areas, models trained on imbalanced datasets may exhibit localized biases that compromise generalizability. Distance-weighted interpolation or extrapolation methods can mitigate this issue by generating representative training datasets that ensure consistent learning across diverse environmental conditions. This enhancement leads to more robust models capable of delivering accurate PM concentration estimates regardless of variations in sensor density.

Investigating how model performance varies with distance from monitoring stations provides crucial insights for optimizing sensor network design. Understanding the rate at which prediction accuracy degrades with increasing distance allows for strategic sensor placement, ensuring optimal spatial coverage. This analysis can also inform the development of correction mechanisms to compensate for potential accuracy losses in regions with lower sensor density. Refining deployment strategies based on these

insights would enable air quality monitoring systems to achieve higher precision in capturing pollutant variations across different urban and industrial landscapes.

Determining the optimal model configuration requires extensive experimentation with different feature sets and neural network architectures. While ANN has demonstrated superior predictive capabilities by effectively integrating geometric and meteorological data, further studies should explore additional feature engineering approaches tailored to specific pollutants and environmental contexts. Comparative evaluations of alternative architectures, including convolutional and recurrent neural networks, could provide deeper insights into the most effective configurations for different air quality monitoring scenarios. The results of such analyses would refine model selection strategies, ensuring that the most suitable algorithms are applied to maximize predictive performance.

A promising direction for future research is the application of Physics-Informed Neural Networks (PINNs) in air quality modeling. Unlike conventional data-driven models, PINNs integrate physical constraints and domain knowledge directly into the learning process, allowing them to capture underlying pollutant dispersion mechanisms more accurately. Given the strong spatial dependencies in air pollution dynamics, incorporating physics-based formulations alongside data-driven learning could lead to more interpretable and generalizable models. A direct comparison between PINNs and ANN under the same experimental framework would provide valuable insights into whether physics-informed approaches enhance performance in complex spatial modeling tasks. This investigation could set a foundation for more advanced hybrid modeling techniques, combining data-driven flexibility with the fundamental principles governing pollutant transport and transformation.

Evaluating model performance across diverse environmental settings, including regions near major emission sources and downwind of industrial zones, ensures robustness and

adaptability. Since air quality varies significantly with local meteorological and geographic factors, validating predictive models across different contexts allows for the identification of strengths and limitations in various conditions. This comparative analysis facilitates the refinement of machine learning models, improving their applicability to real-world air monitoring challenges. Advancing the integration of data-driven and physics-informed methodologies would enhance the accuracy, interpretability, and deployment potential of ML-based air quality prediction systems.

### 5.3 Publication

Kimuya, A. M., Maitethia, D. M., & Kinyua, D. M. (2025). Development of integrated machine learning model for estimation of spatial distribution of particulate matter pollutant in air. *Environmental Research Communications*. Advance online publication. https://doi.org/10.1088/2515-7620/adfb46.

## REFERENCES


Abecasis, L., Gamelas, C. A., Justino, A. R., Dionísio, I., Canha, N., Kertesz, Z., & Almeida, S. M. (2022). Spatial Distribution of Air Pollution, Hotspots and Sources in an Urban-Industrial Area in the Lisbon Metropolitan Area, Portugal—A Biomonitoring Approach. *International Journal of Environmental Research and Public Health*, *19*(3), 1364. https://doi.org/10.3390/ijerph19031364

Abulude, F., Akinnusotu, A., Bello, L., & Feyisetan, A. (2022). *Assessment of AQI, PM10, PM2.5, NO2, O3: The Case of Owo, Nigeria*. *4*, 15–24.

Agramanisti Azdy, R., & Darnis, F. (2020). Use of Haversine Formula in Finding Distance Between Temporary Shelter and Waste End Processing Sites. *Journal of Physics: Conference Series*, *1500*(1), 012104. https://doi.org/10.1088/1742-6596/1500/1/012104

Aldaweesh, S. (2019). *Predicting Hourly Particulate Matter (PM 2.5 ) Concentrations Using Meteorological Data* (p. 140). https://doi.org/10.1109/iCCECE46942.2019.8941696

Aldegunde, J. A. Á., Sánchez, A. F., Saba, M., Bolaños, E. Q., & Palenque, J. Ú. (2022). Analysis of PM2.5 and Meteorological Variables Using Enhanced Geospatial Techniques in Developing Countries: A Case Study of Cartagena de Indias City (Colombia). *Atmosphere*, *13*(4), 506. https://doi.org/10.3390/atmos13040506

Alfano, B., Barretta, L., Del Giudice, A., De Vito, S., Di Francia, G., Esposito, E., Formisano, F., Massera, E., Miglietta, M. L., & Polichetti, T. (2020a). A Review of Low-Cost Particulate Matter Sensors from the Developers' Perspectives. *Sensors*, *20*(23), Article 23. https://doi.org/10.3390/s20236819

Alfano, B., Barretta, L., Del Giudice, A., De Vito, S., Di Francia, G., Esposito, E., Formisano, F., Massera, E., Miglietta, M. L., & Polichetti, T. (2020b). A Review

of Low-Cost Particulate Matter Sensors from the Developers' Perspectives. *Sensors*, *20*(23), Article 23. https://doi.org/10.3390/s20236819

Al-husban, Y. (2021). *Inverse Distance Weighting (IDW) for Estimating Spatial Variation of Monthly and Annually Rainfall in Azraq Basin during the monitor Period (1980-2016)*. *3*.

Badura, M., Batog, P., Drzeniecka-Osiadacz, A., & Modzel, P. (2018). Evaluation of Low-Cost Sensors for Ambient PM $_{2.5}$ Monitoring. *Journal of Sensors*, *2018*, 1–16. https://doi.org/10.1155/2018/5096540

Batmunkh, T., Kim, Y., Jung, J., Park, K., & Tumendemberel, B. (2013). Chemical characteristics of fine particulate matters measured during severe winter haze events in Ulaanbaatar, Mongolia. *Journal of the Air & Waste Management Association*, *63*, 659–670. https://doi.org/10.1080/10962247.2013.776997

Batur Şahin, C., & Diri, B. (2019). Robust Feature Selection With LSTM Recurrent Neural Networks for Artificial Immune Recognition System. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2019.2900118

Behrens, T., Schmidt, K., Viscarra Rossel, R. A., Gries, P., Scholten, T., & MacMillan, R. A. (2018). Spatial modelling with Euclidean distance fields and machine learning. *European Journal of Soil Science*, *69*(5), 757–770. https://doi.org/10.1111/ejss.12687

Bekkar, A., Hssina, B., Douzi, S., & Douzi, K. (2021). Air-pollution prediction in smart city, deep learning approach. *Journal of Big Data*, *8*(1), 161. https://doi.org/10.1186/s40537-021-00548-1

Boldo, E., Medina, S., Le Tertre, A., Hurley, F., Mücke, H.-G., Ballester, F., Aguilera, I., & Eilstein, D. (2006). Apheis: Health Impact Assessment of Long-term Exposure

to PM2.5 in 23 European Cities. *European Journal of Epidemiology*, *21*, 449–458. https://doi.org/10.1007/s10654-006-9014-0

Bronowicka-Mielniczuk, U., Mielniczuk, J., Obroślak, R., & Przystupa, W. (2019). A Comparison of Some Interpolation Techniques for Determining Spatial Distribution of Nitrogen Compounds in Groundwater. *International Journal of Environmental Research*, *13*(4), 679–687. https://doi.org/10.1007/s41742-019-00208-6

Bucek, P., Maršolek, P., & Bílek, J. (2021). Low-Cost Sensors for Air Quality Monitoring—The Current State of the Technology and a Use Overview. *Chemistry-Didactics-Ecology-Metrology*, *26*, 41–54. https://doi.org/10.2478/cdem-2021-0003

Chai, J., Song, J., Zhang, L., Guo, B., & Xu, Y. (2022). Optimization of Land Use Regression Modelling of $PM_{2.5}$ Spatial Variations in Different Seasons across China. *Journal of Sensors*, *2022*, e3659254. https://doi.org/10.1155/2022/3659254

Chen, Y., Cui, S., Chen, P., Yuan, Q., Kang, P., & Zhu, L. (2021). An LSTM-based neural network method of particulate pollution forecast in China. *Environmental Research Letters*, *16*(4), 044006. https://doi.org/10.1088/1748-9326/abe1f5

Cheng, Y., He, K., Du, Z., Zheng, M., Duan, F., & Ma, Y. (2015). Humidity plays an important role in the PM 2.5 pollution in Beijing. *Environmental Pollution*, *197*, 68–75. https://doi.org/10.1016/j.envpol.2014.11.028

Chhajer, P., Shah, M., & Kshirsagar, A. (2022). The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. *Decision Analytics Journal*, *2*, 100015. https://doi.org/10.1016/j.dajour.2021.100015

Choi, K., & Chong, K. (2022). Modified Inverse Distance Weighting Interpolation for Particulate Matter Estimation and Mapping. *Atmosphere*, *13*(5), 846. https://doi.org/10.3390/atmos13050846

Cofre-Martel, S., Lopez Droguett, E., & Modarres, M. (2021). Big Machinery Data Preprocessing Methodology for Data-Driven Models in Prognostics and Health Management. *Sensors*, *21*(20), 6841. https://doi.org/10.3390/s21206841

Constant, N. (2018). Role of Citizen Science in Air Quality Monitoring. In S. M. Charlesworth & C. A. Booth (Eds.), *Urban Pollution* (1st ed., pp. 303–312). Wiley. https://doi.org/10.1002/9781119260493.ch23

Crinnion, W. (2017). Particulate Matter Is a Surprisingly Common Contributor to Disease. *Integrative Medicine: A Clinician's Journal*, *16*(4), 8–12.

Davies, T. M., Banerjee, S., Martin, A. P., & Turnbull, R. E. (2022). A Nearest-Neighbour Gaussian Process Spatial Factor Model for Censored, Multi-Depth Geochemical Data. *Journal of the Royal Statistical Society Series C: Applied Statistics*, *71*(4), 1014–1043. https://doi.org/10.1111/rssc.12565

Denny, Y., Spits Warnars, H. L. H., Budiharto, W., Kistijantoro, A. I., Heryadi, Y., & Lukas, L. (2018). *Lstm And Simple Rnn Comparison In The Problem Of Sequence To Sequence On Conversation Data Using Bahasa Indonesia* (p. 56). https://doi.org/10.1109/INAPR.2018.8627029

Di Bucchianico, A. (2007). Coefficient of Determination ( $R^2$ ). In F. Ruggeri, R. S. Kenett, & F. W. Faltin (Eds.), *Encyclopedia of Statistics in Quality and Reliability* (1st ed.). Wiley. https://doi.org/10.1002/9780470061572.eqr173

Duan, J., Gong, Y., Luo, J., & Zhao, Z. (2023). Air-quality prediction based on the ARIMA-CNN-LSTM combination model optimized by dung beetle optimizer. *Scientific Reports*, *13*(1), Article 1. https://doi.org/10.1038/s41598-023-36620-4

Dubey, R., Patra, A. K., Joshi, J., Blankenberg, D., Kolluru, S. S. R., Madhu, B., & Raval, S. (2022). Evaluation of low-cost particulate matter sensors OPC N2 and PM Nova for aerosol monitoring. *Atmospheric Pollution Research*, *13*(3), 101335. https://doi.org/10.1016/j.apr.2022.101335

Duffney, P. F., Stanek, L. W., & Brown, J. S. (2023). Air pollution: Sources, regulation, and health effects. In *Reference Module in Biomedical Sciences*. Elsevier. https://doi.org/10.1016/B978-0-12-824315-2.00754-5

Ebrahimi-Khusfi, Z., Taghizadeh-Mehrjardi, R., Kazemi, M., & Nafarzadegan, A. R. (2021). Predicting the ground-level pollutants concentrations and identifying the influencing factors using machine learning, wavelet transformation, and remote sensing techniques. *Atmospheric Pollution Research*, *12*(5), 101064. https://doi.org/10.1016/j.apr.2021.101064

Faour, A., Abboud, M., Germanos, G., & Farah, W. (2022). Assessment of the exposure to PM2.5 in different Lebanese microenvironments at different temporal scales. *Environmental Monitoring and Assessment*, *195*(1), 21. https://doi.org/10.1007/s10661-022-10607-6

Ganguli, R. (2002). NOISE AND OUTLIER REMOVAL FROM JET ENGINE HEALTH SIGNALS USING WEIGHTED FIR MEDIAN HYBRID FILTERS. *Mechanical Systems and Signal Processing*, *16*(6), 967–978. https://doi.org/10.1006/mssp.2002.1477

Giordano, M. R., Malings, C., Pandis, S. N., Presto, A. A., McNeill, V. F., Westervelt, D. M., Beekmann, M., & Subramanian, R. (2021). From low-cost sensors to high-quality data: A summary of challenges and best practices for effectively calibrating low-cost particulate matter mass sensors. *Journal of Aerosol Science*, *158*, 105833. https://doi.org/10.1016/j.jaerosci.2021.105833

Gong, W., Zhang, T., Zhu, Z., Ma, Y., Ma, X., & Wang, W. (2015). Characteristics of PM1.0, PM2.5, and PM10, and Their Relation to Black Carbon in Wuhan, Central China. *Atmosphere*, *6*(9), 1377–1387. https://doi.org/10.3390/atmos6091377

Goulier, L., Paas, B., Ehrnsperger, L., & Klemm, O. (2020). Modelling of Urban Air Pollutant Concentrations with Artificial Neural Networks Using Novel Input Variables. *International Journal of Environmental Research and Public Health*, *17*(6), 2025. https://doi.org/10.3390/ijerph17062025

Guiñón, J., Ortega, E., García-Antón, J., & Pérez-Herranz, V. (2007). *Moving average and Savitzki-Golay smoothing filters using Mathcad*.

Guo, Q., Ren, M., Wu, S., Sun, Y., Wang, J., Wang, Q., Ma, Y., Song, X., & Chen, Y. (2022a). Applications of artificial intelligence in the field of air pollution: A bibliometric analysis. *Frontiers in Public Health*, *10*. https://www.frontiersin.org/articles/10.3389/fpubh.2022.933665

Guo, Q., Ren, M., Wu, S., Sun, Y., Wang, J., Wang, Q., Ma, Y., Song, X., & Chen, Y. (2022b). Applications of artificial intelligence in the field of air pollution: A bibliometric analysis. *Frontiers in Public Health*, *10*, 933665. https://doi.org/10.3389/fpubh.2022.933665

Hagan, D., Lewis, A., von Schneidemesser, E., Peltier, R., Lung, S., Jones, R., Zellweger, C., Karppinen, A., Penza, M., Dye, T., Huglin, C., Ning, Z., Leigh, R., Laurent, O., Carmichael, G., Beig, G., Cohen, R., Cross, E., Gentner, D., & Tarasova, O. (2018). *Low-cost sensors for the measurement of atmospheric composition: Overview of topic and future applications*.

Han, P., Mei, H., Liu, D., Zeng, N., Tang, X., Wang, Y., & Pan, Y. (2021). Calibrations of Low-Cost Air Pollution Monitoring Sensors for CO, NO2, O3, and SO2. *Sensors*, *21*, 256. https://doi.org/10.3390/s21010256

Hart, R., Liang, L., & Dong, P. (2020). Monitoring, Mapping, and Modeling Spatial–Temporal Patterns of PM2.5 for Improved Understanding of Air Pollution Dynamics Using Portable Sensing Technologies. *International Journal of Environmental Research and Public Health*, *17*(14), 4914. https://doi.org/10.3390/ijerph17144914

Hawkins, D. (2004). The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences*, *44*, 1–12. https://doi.org/10.1021/ci0342472

He, M., Kuerbanjiang, N., & Dhaniyala, S. (2019). Performance characteristics of the low-cost Plantower PMS optical sensor. *Aerosol Science and Technology*, *54*, 1–11. https://doi.org/10.1080/02786826.2019.1696015

Hidy, G. M. (2019). Atmospheric Chemistry in a Box or a Bag. *Atmosphere*, *10*(7), Article 7. https://doi.org/10.3390/atmos10070401

Hu, Z., Liebens, J., & Rao, K. R. (2011). Merging Satellite Measurement with Ground-Based Air Quality Monitoring Data to Assess Health Effects of Fine Particulate Matter Pollution. In J. A. Maantay & S. McLafferty (Eds.), *Geospatial Analysis of Environmental Health* (pp. 395–409). Springer Netherlands. https://doi.org/10.1007/978-94-007-0329-2_20

Humans, I. W. G. on the E. of C. R. to. (2016). Sources of air pollutants. In *Outdoor air pollution*. International Agency for Research on Cancer. https://www.ncbi.nlm.nih.gov/books/NBK368029/

Huy, L. (2015). *Evaluation of performance of photochemical smog modelling system for air quality management in Vietnam*.

Hyndman, R. (2010). Moving Averages. In *International Encyclopedia of Statistical Science* (pp. 866–869). https://doi.org/10.1007/978-3-642-04898-2_380

Javed, W., & Guo, B. (2021). Performance Evaluation of Real-time DustTrak Monitors for Outdoor Particulate Mass Measurements in a Desert Environment. *Aerosol and Air Quality Research*, *21*. https://doi.org/10.4209/aaqr.200631

Jiang, X., Luo, Y., & Zhang, B. (2021). Prediction of PM2.5 Concentration Based on the LSTM-TSLightGBM Variable Weight Combination Model. *Atmosphere*, *12*(9), Article 9. https://doi.org/10.3390/atmos12091211

Jones, S. G., Ashby, A. J., Momin, S. R., & Naidoo, A. (2010). Spatial Implications Associated with Using Euclidean Distance Measurements and Geographic Centroid Imputation in Health Care Research. *Health Services Research*, *45*(1), 316–327. https://doi.org/10.1111/j.1475-6773.2009.01044.x

Karagulian, F., Barbiere, M., Kotsev, A., Spinelle, L., Gerboles, M., Lagler, F., Redon, N., Crunaire, S., & Borowiak, A. (2019). Review of the Performance of Low-Cost Sensors for Air Quality Monitoring. *Atmosphere*, *10*(9), 506. https://doi.org/10.3390/atmos10090506

Kelly, F. J., & Fussell, J. C. (2015). Air pollution and public health: Emerging hazards and improved understanding of risk. *Environmental Geochemistry and Health*, *37*(4), 631–649. https://doi.org/10.1007/s10653-015-9720-1

Khreis, H., Johnson, J., Jack, K., Dadashova, B., & Park, E. S. (2022). Evaluating the Performance of Low-Cost Air Quality Monitors in Dallas, Texas. *International Journal of Environmental Research and Public Health*, *19*(3), 1647. https://doi.org/10.3390/ijerph19031647

Kim, B.-Y., Lim, Y.-K., & Cha, J. W. (2022). Short-term prediction of particulate matter (PM10 and PM2.5) in Seoul, South Korea using tree-based machine learning

algorithms. *Atmospheric Pollution Research*, *13*(10), 101547. https://doi.org/10.1016/j.apr.2022.101547

Kim, K. H., Lee, S.-B., Woo, D., & Bae, G.-N. (2015). Influence of wind direction and speed on the transport of particle-bound PAHs in a roadway environment. *Atmospheric Pollution Research*, *6*(6), 1024–1034. https://doi.org/10.1016/j.apr.2015.05.007

Kleijnen, J. (2017). Kriging: Methods and Applications. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3075151

Kodzius, R., Sabir, D., Rabiei, N., Fattah, A., Wang, X., Liu, J., Gong, X., & Damiati, S. (2018). *The Pollutant Particle Size and Chemistry Matters*. https://doi.org/10.20944/preprints201805.0004.v1

Kolluri, J., Kotte, V., Phridviraj, M. S. B., & Razia, S. (2020). *Reducing Overfitting Problem in Machine Learning Using Novel L1/4 Regularization Method* (p. 938). https://doi.org/10.1109/ICOEI48184.2020.9142992

Kshirsagar, M. P., & Khare, K. C. (2023). Support Vector Regression Models of Stormwater Quality for a Mixed Urban Land Use. *Hydrology*, *10*(3), Article 3. https://doi.org/10.3390/hydrology10030066

Kumar, A., & Gurjar, B. R. (2019). Low-Cost Sensors for Air Quality Monitoring in Developing Countries – A Critical View. *Asian Journal of Water, Environment and Pollution*, *16*(2), 65–70. https://doi.org/10.3233/AJW190021

Kumar, P., Morawska, L., Martani, C., Biskos, G., Neophytou, M., Di Sabatino, S., Bell, M., Norford, L., & Britter, R. (2015). The rise of low-cost sensing for managing air pollution in cities. *Environment International*, *75*, 199–205. https://doi.org/10.1016/j.envint.2014.11.019

Lakowicz, J. R. (Ed.). (2006). *Principles of Fluorescence Spectroscopy*. Springer US. https://doi.org/10.1007/978-0-387-46312-4

Larson, P. S., Espira, L., Glenn, B. E., Larson, M. C., Crowe, C. S., Jang, S., & O'Neill, M. S. (2022). Long-Term PM2.5 Exposure Is Associated with Symptoms of Acute Respiratory Infections among Children under Five Years of Age in Kenya, 2014. *International Journal of Environmental Research and Public Health*, *19*(5), 2525. https://doi.org/10.3390/ijerph19052525

Li, C., Dai, Z., Yang, L., & Ma, Z. (2019). Spatiotemporal Characteristics of Air Quality across Weifang from 2014–2018. *International Journal of Environmental Research and Public Health*, *16*(17), 3122. https://doi.org/10.3390/ijerph16173122

Li, H., Ge, M., & Zhang, M. (2022). Spatio-temporal distribution of tuberculosis and the effects of environmental factors in China. *BMC Infectious Diseases*, *22*(1), 565. https://doi.org/10.1186/s12879-022-07539-4

Li, L., Gong, J., & Zhou, J. (2014). Spatial Interpolation of Fine Particulate Matter Concentrations Using the Shortest Wind-Field Path Distance. *PLoS ONE*, *9*(5), e96111. https://doi.org/10.1371/journal.pone.0096111

Li, L., Tian, J., Zhang, X., Holt, J., & Piltner, R. (2015). Estimating Population Exposure to Fine Particulate Matter in the Conterminous U.S. using Shape Function-based Spatiotemporal Interpolation Method: A County Level Analysis. *GSTF International Journal on Computing*, *1*, 24–30.

Li, L., Zhou, X., Kalo, M., & Piltner, R. (2016). Spatiotemporal Interpolation Methods for the Application of Estimating Population Exposure to Fine Particulate Matter in the Contiguous U.S. and a Real-Time Web Application. *International Journal*

*of Environmental Research and Public Health*, *13*(8), 749. https://doi.org/10.3390/ijerph13080749

Li, X., Ma, Y., Wang, Y., Liu, N., & Hong, Y. (2017). Temporal and spatial analyses of particulate matter (PM 10 and PM 2.5 ) and its relationship with meteorological parameters over an urban city in northeast China. *Atmospheric Research*, *198*. https://doi.org/10.1016/j.atmosres.2017.08.023

Liang, Y.-C., Maimury, Y., Chen, A. H.-L., & Juarez, J. R. C. (2020). Machine Learning-Based Prediction of Air Quality. *Applied Sciences*, *10*(24), Article 24. https://doi.org/10.3390/app10249151

Lin, J., Lin, Y., Lin, S., & Dong, J. (2023). The characteristic of atmospheric particulate matter and the influence factors in Xiamen for air quality management. *Frontiers in Environmental Science*, *11*, 1220720. https://doi.org/10.3389/fenvs.2023.1220720

Liu, Y., Wang, Y., & Zhang, J. (2012). New Machine Learning Algorithm: Random Forest. In B. Liu, M. Ma, & J. Chang (Eds.), *Information Computing and Applications* (Vol. 7473, pp. 246–252). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-34062-8_32

Lutman, E., Jones, S., Hill, R. A., McDonald, P., & Lambers, B. (2004). Comparison between the predictions of a Gaussian plume model and a Lagrangian particle dispersion model for annual average calculations of long-range dispersion of radionuclides. *Journal of Environmental Radioactivity*, *75*, 339–355. https://doi.org/10.1016/j.jenvrad.2003.11.013

Mahajan, S., Kumar, P., Pinto, J. A., Riccetti, A., Schaaf, K., Camprodon, G., Smári, V., Passani, A., & Forino, G. (2020). A citizen science approach for enhancing

public understanding of air pollution. *Sustainable Cities and Society*, *52*, 101800. https://doi.org/10.1016/j.scs.2019.101800

Maleika, W. (2020). Inverse distance weighting method optimization in the process of digital terrain model creation based on data collected from a multibeam echosounder. *Applied Geomatics*, *12*(4), 397–407. https://doi.org/10.1007/s12518-020-00307-6

Manisalidis, I., Stavropoulou, E., Stavropoulos, A., & Bezirtzoglou, E. (2020a). Environmental and Health Impacts of Air Pollution: A Review. *Frontiers in Public Health*, *8*, 14. https://doi.org/10.3389/fpubh.2020.00014

Manisalidis, I., Stavropoulou, E., Stavropoulos, A., & Bezirtzoglou, E. (2020b). Environmental and Health Impacts of Air Pollution: A Review. *Frontiers in Public Health*, *8*. https://www.frontiersin.org/articles/10.3389/fpubh.2020.00014

Martin, R. V. (2008). Satellite remote sensing of surface air quality. *Atmospheric Environment*, *42*(34), 7823–7843. https://doi.org/10.1016/j.atmosenv.2008.07.018

Mehmood, K., Bao, Y., Saifullah, Cheng, W., Khan, M. A., Siddique, N., Abrar, M. M., Soban, A., Fahad, S., & Naidu, R. (2022). Predicting the quality of air with machine learning approaches: Current research priorities and future perspectives. *Journal of Cleaner Production*, *379*, 134656. https://doi.org/10.1016/j.jclepro.2022.134656

Méndez, M., Merayo, M. G., & Núñez, M. (2023). Machine learning algorithms to forecast air quality: A survey. *Artificial Intelligence Review*, *56*(9), 10031–10066. https://doi.org/10.1007/s10462-023-10424-4

Meng, M., Cao, S.-J., Kumar, P., Tang, X., & Zhuangbo, F. (2021). Spatial distribution characteristics of PM2.5 concentration around residential buildings in urban

traffic-intensive areas: From the perspectives of health and safety. *Safety Science*. https://doi.org/10.1016/j.enbuild.2021.105318

Molina Rueda, E., Carter, E., L'Orange, C., Quinn, C., & Volckens, J. (2023). Size-Resolved Field Performance of Low-Cost Sensors for Particulate Matter Air Pollution. *Environmental Science & Technology Letters*, *10*(3), 247–253. https://doi.org/10.1021/acs.estlett.3c00030

Mukherjee, A., & Agrawal, M. (2018). A Global Perspective of Fine Particulate Matter Pollution and Its Health Effects. *Reviews of Environmental Contamination and Toxicology*, *244*, 5–51. https://doi.org/10.1007/398_2017_3

Munir, S., Mayfield, M., Coca, D., & Jubb, S. A. (2019). Structuring an integrated air quality monitoring network in large urban areas – Discussing the purpose, criteria and deployment strategy. *Atmospheric Environment: X*, *2*, 100027. https://doi.org/10.1016/j.aeaoa.2019.100027

Munyati, C., & Sinthumule, N. I. (2021). Comparative suitability of ordinary kriging and Inverse Distance Weighted interpolation for indicating intactness gradients on threatened savannah woodland and forest stands. *Environmental and Sustainability Indicators*, *12*, 100151. https://doi.org/10.1016/j.indic.2021.100151

Narayana, M. V., Jalihal, D., & Nagendra, S. M. S. (2022). Establishing A Sustainable Low-Cost Air Quality Monitoring Setup: A Survey of the State-of-the-Art. *Sensors (Basel, Switzerland)*, *22*(1), 394. https://doi.org/10.3390/s22010394

O'Connor, S., O'Connor, P. F., Feng, H. A., & Ashley, K. (2014). Gravimetric Analysis of Particulate Matter using Air Samplers Housing Internal Filtration Capsules. *Gefahrstoffe, Reinhaltung Der Luft = Air Quality Control / Herausgeber, BIA Und KRdL Im VDI Und DIN*, *74*(10), 403–410.

Pantusheva, M., Mitkov, R., Hristov, P. O., & Petrova-Antonova, D. (2022). Air Pollution Dispersion Modelling in Urban Environment Using CFD: A Systematic Review. *Atmosphere*, *13*(10), Article 10. https://doi.org/10.3390/atmos13101640

Pawul, M., & Śliwka, M. (2016). Application of artificial neural networks for prediction of air pollution levels in environmental monitoring. *Journal of Ecological Engineering*, *17*, 190–196. https://doi.org/10.12911/22998993/64828

Peralta, B., Sepúlveda, T., Nicolis, O., & Caro, L. (2022). Space-Time Prediction of PM2.5 Concentrations in Santiago de Chile Using LSTM Networks. *Applied Sciences*, *12*(22), 11317. https://doi.org/10.3390/app122211317

Plocoste, T., & Laventure, S. (2023). Forecasting PM10 Concentrations in the Caribbean Area Using Machine Learning Models. *Atmosphere*, *14*(1), Article 1. https://doi.org/10.3390/atmos14010134

*PMS5003 Series Manual by Adafruit Industries LLC Datasheet | DigiKey*. (n.d.). Retrieved June 18, 2024, from https://www.digikey.com/en/htmldatasheets/production/2903006/0/0/1/pms5003-series-manual

Qiao, Z., Cui, S., Pei, C., Ye, Z., Wu, X., Lei, L., Luo, T., Zhang, Z., Li, X., & Zhu, W. (2022). Regional Predictions of Air Pollution in Guangzhou: Preliminary Results and Multi-Model Cross-Validations. *Atmosphere*, *13*(10), Article 10. https://doi.org/10.3390/atmos13101527

Raja, S., R Chandrasekaran, S., Lin, L., Xia, X., Hopke, P., & Valsaraj, K. (2016). Analysis of Beta Attenuation Monitor Filter Rolls for Particulate Matter Speciation. *Aerosol and Air Quality Research*, *17*. https://doi.org/10.4209/aaqr.2016.03.0122

rm, N. (2019). *A Comparative Study on Air Pollution Modelling Techniques – A Review*.

Rohajawati, S., Setyodewi, H., Tresnanto, F. M. A., Marianthi, D., & Sihotang, M. T. B. (2024). KNOWLEDGE MANAGEMENT APPROACH IN COMPARATIVE STUDY OF AIR POLLUTION PREDICTION MODEL. *Applied Computer Science*, *20*(1), 173–188. https://doi.org/10.35784/acs-2024-11

Rosenberg, P., Dean, A., Williams, P., Dorsey, J., Minikin, A., Pickering, M., & Petzold, A. (2012). Particle sizing calibration with refractive index correction for light scattering optical particle counters and impacts upon PCASP and CDP data collected during the Fennec campaign. *Atmos. Meas. Tech.*, *5*, 1147–1163. https://doi.org/10.5194/amt-5-1147-2012

Rybarczyk, Y., & Zalakeviciute, R. (2018). Machine Learning Approaches for Outdoor Air Quality Modelling: A Systematic Review. *Applied Sciences*, *8*, 2570. https://doi.org/10.3390/app8122570

Sabir, M. A., Nawaz, M. F., Khan, T. H., Zulfiqar, U., Haider, F. U., Rehman, A., Ahmad, I., Rasheed, F., Gul, S., Hussain, S., Iqbal, R., Chaudhary, T., Mustafa, A. E.-Z. M. A., & Elshikh, M. S. (2024). Investigating seasonal air quality variations consequent to the urban vegetation in the metropolis of Faisalabad, Pakistan. *Scientific Reports*, *14*(1), 452. https://doi.org/10.1038/s41598-023-47512-y

Sadrizadeh, S., Yao, R., Yuan, F., Awbi, H., Bahnfleth, W., Bi, Y., Cao, G., Croitoru, C., de Dear, R., Haghighat, F., Kumar, P., Malayeri, M., Nasiri, F., Ruud, M., Sadeghian, P., Wargocki, P., Xiong, J., Yu, W., & Li, B. (2022). Indoor air quality and health in schools: A critical review for developing the roadmap for the future school environment. *Journal of Building Engineering*, *57*, 104908. https://doi.org/10.1016/j.jobe.2022.104908

Saiohai, J., Bualert, S., Thongyen, T., Duangmal, K., Choomanee, P., & Szymanski, W. W. (2023). Statistical PM2.5 Prediction in an Urban Area Using Vertical

Meteorological Factors. *Atmosphere*, *14*(3), Article 3. https://doi.org/10.3390/atmos14030589

Sajid, A. H., Rudra, R., & Parkin, G. (2013). Systematic Evaluation of Kriging and Inverse Distance Weighting Methods for Spatial Analysis of Soil Bulk Density. *Canadian Biosystems Engineering*, *55*, 1.1-1.13. https://doi.org/10.7451/CBE.2013.55.1

Sakhrieh, A., Hamdan, M., & Ata, M. (2021). Air Quality Assessment and Forecasting Using Neural Network Model. *Journal of Ecological Engineering*, *22*, 1–11. https://doi.org/10.12911/22998993/137444

Sánchez, A., Garcia Nieto, P. J., Riesgo Fernández, P., Díaz, J. J., & Iglesias-Rodriguez, F.-J. (2011). Application of a SVM-based regression model to the air quality study at local scale in the Avilés urban area (Spain). *Mathematical and Computer Modelling*, *54*, 1453–1466. https://doi.org/10.1016/j.mcm.2011.04.017

Saragih, T. H., & Mazdadi, M. I. (2023). Comparison of Air Quality Prediction using Random Forest and Gradient Boosting Tree. *2023 Eighth International Conference on Informatics and Computing (ICIC)*, 1–5. https://doi.org/10.1109/ICIC60109.2023.10382104

Schraufnagel, D. E. (2020). The health effects of ultrafine particles. *Experimental & Molecular Medicine*, *52*(3), 311–317. https://doi.org/10.1038/s12276-020-0403-3

Sharma, M., Jain, S., Mittal, S., & Sheakh, T. (2021). Forecasting And Prediction Of Air Pollutants Concentrates Using Machine Learning Techniques: The Case Of India. *IOP Conference Series: Materials Science and Engineering*, *1022*, 012123. https://doi.org/10.1088/1757-899X/1022/1/012123

Shiferaw, A. B., Kumie, A., & Tefera, W. (2023). The spatial and temporal variation of fine particulate matter pollution in Ethiopia: Data from the Atmospheric

Composition Analysis Group (1998–2019). *PLOS ONE*, *18*(3), e0283457. https://doi.org/10.1371/journal.pone.0283457

Siahaan, A. P. U. (2017). *Haversine Method in Looking for the Nearest Masjid* [Preprint]. INA-Rxiv. https://doi.org/10.31227/osf.io/eb3ja

Sillmann, J., Thorarinsdottir, T., Keenlyside, N., Schaller, N., Alexander, L. V., Hegerl, G., Seneviratne, S. I., Vautard, R., Zhang, X., & Zwiers, F. W. (2017). Understanding, modeling and predicting weather and climate extremes: Challenges and opportunities. *Weather and Climate Extremes*, *18*, 65–74. https://doi.org/10.1016/j.wace.2017.10.003

Stanek, L. W., & Brown, J. S. (2019). Air Pollution: Sources, Regulation, and Health Effects. In *Reference Module in Biomedical Sciences* (p. B9780128012383113844). Elsevier. https://doi.org/10.1016/B978-0-12-801238-3.11384-4

Sun, R., Zhou, Y., Wu, J., & Gong, Z. (2019). Influencing Factors of PM2.5 Pollution: Disaster Points of Meteorological Factors. *International Journal of Environmental Research and Public Health*, *16*(20), 3891. https://doi.org/10.3390/ijerph16203891

Tai, A., Mickley, L., Jacob, D., Leibensperger, E., Zhang, L., & Fisher, J. (2011). Meteorological modes of variability for fine particulate matter (PM2.5) air quality in the United States: Implications for PM2.5 sensitivity to climate change. *AGU Fall Meeting Abstracts*, *12*, 05. https://doi.org/10.5194/acp-12-3131-2012

Tella, A., Balogun, A.-L., Adebisi, N., & Abdullah, S. (2021). Spatial assessment of PM10 hotspots using Random Forest, K-Nearest Neighbour and Naïve Bayes. *Atmospheric Pollution Research*, *12*(10), 101202. https://doi.org/10.1016/j.apr.2021.101202

Thangavel, P., Park, D., & Lee, Y.-C. (2022). Recent Insights into Particulate Matter (PM2.5)-Mediated Toxicity in Humans: An Overview. *International Journal of Environmental Research and Public Health*, *19*(12), 7511. https://doi.org/10.3390/ijerph19127511

Ukaogo, P. O., Ewuzie, U., & Onwuka, C. V. (2020). 21 - Environmental pollution: Causes, effects, and the remedies. In P. Chowdhary, A. Raj, D. Verma, & Y. Akhter (Eds.), *Microorganisms for Sustainable Environment and Health* (pp. 419–429). Elsevier. https://doi.org/10.1016/B978-0-12-819001-2.00021-8

Ventura, L., Lima, I., & Luna, A. (2013). *Influence of meteorological parameters on air quality*. 3256.

Wallace, L., & Hopke, P. K. (2022). Measuring Particle Concentrations and Composition in Indoor Air. In Y. Zhang, P. K. Hopke, & C. Mandin (Eds.), *Handbook of Indoor Air Quality* (pp. 517–567). Springer Nature. https://doi.org/10.1007/978-981-16-7680-2_19

Wang, Y., Liu, K., He, Y., Wang, P., Chen, Y., Xue, H., Huang, C., & Li, L. (2024). Enhancing Air Quality Forecasting: A Novel Spatio-Temporal Model Integrating Graph Convolution and Multi-Head Attention Mechanism. *Atmosphere*, *15*(4), 418. https://doi.org/10.3390/atmos15040418

Wang, Y., Yang, W., Han, B., Zhang, W., Chen, M., & Bai, Z. (2016). Gravimetric analysis for PM2.5 mass concentration based on year-round monitoring at an urban site in Beijing. *Journal of Environmental Sciences*, *40*, 154–160. https://doi.org/10.1016/j.jes.2015.09.015

Won, W.-S., Oh, R., Lee, W., Ku, S., Su, P.-C., & Yoon, Y.-J. (2021). Hygroscopic properties of particulate matter and effects of their interactions with weather on

visibility. *Scientific Reports*, *11*(1), 16401. https://doi.org/10.1038/s41598-021-95834-6

Wong, P.-Y., Lee, H.-Y., Chen, Y.-C., Zeng, Y.-T., Chern, Y.-R., Chen, N.-T., Candice Lung, S.-C., Su, H.-J., & Wu, C.-D. (2021). Using a land use regression model with machine learning to estimate ground level PM2.5. *Environmental Pollution*, *277*, 116846. https://doi.org/10.1016/j.envpol.2021.116846

Wu, Y., & Xue, W. (2024). Data-Driven Weather Forecasting and Climate Modeling from the Perspective of Development. *Atmosphere*, *15*(6), 689. https://doi.org/10.3390/atmos15060689

Xiao, F., Yang, M., Fan, H., Fan, G., & Al-qaness, M. A. A. (2020). An improved deep learning model for predicting daily PM2.5 concentration. *Scientific Reports*, *10*(1), Article 1. https://doi.org/10.1038/s41598-020-77757-w

Xiao, Q., Chang, H. H., Geng, G., & Liu, Y. (2018). An Ensemble Machine-Learning Model To Predict Historical PM2.5 Concentrations in China from Satellite Data. *Environmental Science & Technology*, *52*(22), 13260–13269. https://doi.org/10.1021/acs.est.8b02917

Xing, Y.-F., Xu, Y.-H., Shi, M.-H., & Lian, Y.-X. (2016). The impact of PM2.5 on the human respiratory system. *Journal of Thoracic Disease*, *8*(1), E69–E74. https://doi.org/10.3978/j.issn.2072-1439.2016.01.19

Xu, S., Li, W., Zhu, Y., & Xu, A. (2022). A novel hybrid model for six main pollutant concentrations forecasting based on improved LSTM neural networks. *Scientific Reports*, *12*(1), Article 1. https://doi.org/10.1038/s41598-022-17754-3

Xu, X., & Zhang, C. (2020). Estimation of ground-level PM2.5 concentration using MODIS AOD and corrected regression model over Beijing, China. *PLOS ONE*, *15*, e0240430. https://doi.org/10.1371/journal.pone.0240430

Yu, M., Zhang, S., Zhang, K., Yin, J., Varela, M., & Miao, J. (2023). Developing high-resolution PM2.5 exposure models by integrating low-cost sensors, automated machine learning, and big human mobility data. *Frontiers in Environmental Science*, *11*. https://www.frontiersin.org/articles/10.3389/fenvs.2023.1223160

Yu, Y., Kwok, K. C. S., Liu, X. P., & Zhang, Y. (2017). Air pollutant dispersion around high-rise buildings under different angles of wind incidence. *Journal of Wind Engineering and Industrial Aerodynamics*, *167*, 51–61. https://doi.org/10.1016/j.jweia.2017.04.006

Yu, Y., Yang, W., Wang, B., & Meyers, M. A. (2017). Structure and mechanical behavior of human hair. *Materials Science and Engineering: C*, *73*, 152–163. https://doi.org/10.1016/j.msec.2016.12.008

Zaman, N. A. F. K., Kanniah, K. D., Kaskaoutis, D. G., & Latif, M. T. (2021). Evaluation of Machine Learning Models for Estimating PM2.5 Concentrations across Malaysia. *Applied Sciences*, *11*(16), Article 16. https://doi.org/10.3390/app11167326

Zamani Joharestani, M., Cao, C., Ni, X., Bashir, B., & Talebiesfandarani, S. (2019). PM2.5 Prediction Based on Random Forest, XGBoost, and Deep Learning Using Multisource Remote Sensing Data. *Atmosphere*, *10*(7), Article 7. https://doi.org/10.3390/atmos10070373

Zang, L., Wang, Z., Zhu, B., & Zhang, Y. (2019). Roles of Relative Humidity in Aerosol Pollution Aggravation over Central China during Wintertime. *International Journal of Environmental Research and Public Health*, *16*(22), 4422. https://doi.org/10.3390/ijerph16224422

Zhang, G., Rui, X., & Fan, Y. (2018a). Critical Review of Methods to Estimate PM2.5 Concentrations within Specified Research Region. *ISPRS International Journal of Geo-Information*, *7*(9), Article 9. https://doi.org/10.3390/ijgi7090368

Zhang, G., Rui, X., & Fan, Y. (2018b). Critical Review of Methods to Estimate PM2.5 Concentrations within Specified Research Region. *ISPRS International Journal of Geo-Information*, *7*(9), Article 9. https://doi.org/10.3390/ijgi7090368

Zhang, S., Shen, X., Sun, J., Zhang, Y., Zhang, X., Xia, C., Hu, X., Zhong, J., Wang, J., & Liu, S. (2023). Atmospheric Particle Hygroscopicity and the Influence by Oxidation State of Organic Aerosols in Urban Beijing. *Journal of Environmental Sciences*, *124*, 544–556. https://doi.org/10.1016/j.jes.2021.11.019

Zhang, Z. (2018). *Artificial Neural Network* (pp. 1–35). https://doi.org/10.1007/978-3-319-67340-0_1

Zickus, M., Greig, A. J., & Niranjan, M. (2002). Comparison of Four Machine Learning Methods for Predicting PM10 Concentrations in Helsinki, Finland. *Water Air and Soil Pollution Focus*, *2*, 717–729. https://doi.org/10.1023/A:1021321820639

# APPENDICES

The analysis presented through Appendix I to Appendix IV was conducted on Google Colab, leveraging Python's robust ecosystem for machine learning development. The implementation utilized key libraries including Keras for neural network architectures (ANNs, LSTM), scikit-learn for Support Vector Regression (SVR) and Random Forest models, and pandas for data manipulation. Preprocessing tasks such as noise reduction via median filtering and categorical encoding were executed using NumPy and scikit-learn, while model performance was assessed through standardized metrics ($R^2$, RMSE, MAE) and visualized with Matplotlib. The integrated workflow, from data ingestion to model persistence with joblib and TensorFlow, underscores a reproducible, cloud-based computational environment tailored for scalable machine learning experimentation and validation.

# Appendix I: Implementation of the ANNs PM Model

```
# Import necessary libraries
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
import matplotlib.pyplot as plt

# Define function for 1D median filtering
def median_filter_1d(signal):
    """Performs median filtering on a 1D signal."""
    filtered_signal = np.empty_like(signal)
    for i in range(signal.shape[0]):
        median = np.median(signal[max(0, i-1):min(signal.shape[0], i+2)])
        filtered_signal[i] = median
    return filtered_signal

# Load dataset
data = pd.read_csv('PM_Training_DataSet_2024.csv')

# Apply median filtering to PM columns
filtered_data = data.copy()
for column in ['PM1.0', 'PM2.5', 'PM10']:
    filtered_data[column] = median_filter_1d(data[column].values)

# Save filtered dataset
filtered_data.to_csv('PM_Filtered_Training_Dataset.csv', index=False)
print("Filtered data saved as 'PM_Filtered_Training_Dataset.csv'.")

# Read filtered dataset
filtered_data = pd.read_csv('PM_Filtered_Training_Dataset.csv')

# Encode categorical variable 'Orientation' using LabelEncoder
le = LabelEncoder()
filtered_data['Orientation'] = le.fit_transform(filtered_data['Orientation'])

# Define timesteps
timesteps = 3

# Define feature combinations
feature_combinations = [
    ['Euclidean_D', 'Orientation'],
    ['Wind speed', 'Euclidean_D', 'Orientation'],
    ['Temperature', 'Euclidean_D', 'Orientation'],
    ['Humidity', 'Euclidean_D', 'Orientation'],
    ['Temperature', 'Humidity', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Temperature', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Humidity', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Temperature', 'Humidity', 'Euclidean_D', 'Orientation']
]

# Define parameters and epochs for each PM type
parameters = {
    'PM1.0': {'epochs': 30},
    'PM2.5': {'epochs': 40},
    'PM10': {'epochs': 35}
}

# Loop through each parameter and feature combination
for parameter in parameters:
    for features in feature_combinations:
        feature_columns = features + [parameter]
```

```
    parameter_index = filtered_data.columns.get_loc(parameter)

    # Split data into train, validation, and test sets
    train_size = int(len(filtered_data) * 0.8)
    val_size = int(len(filtered_data) * 0.1)
    train_data = filtered_data.iloc[:train_size, :].values
    val_data = filtered_data.iloc[train_size:train_size+val_size, :].values
    test_data = filtered_data.iloc[train_size+val_size:, :].values

    # Prepare train, validation, and test data with timesteps
    X_train, y_train = [], []
    for i in range(timesteps, len(train_data)):
        X_train.append(train_data[i - timesteps:i, filtered_data.columns.get_indexer(feature_columns)])
        y_train.append(train_data[i, parameter_index])
    X_train, y_train = np.array(X_train), np.array(y_train)

    # Repeat for validation set
    X_val, y_val = [], []
    for i in range(timesteps, len(val_data)):
        X_val.append(val_data[i - timesteps:i, filtered_data.columns.get_indexer(feature_columns)])
        y_val.append(val_data[i, parameter_index])
    X_val, y_val = np.array(X_val), np.array(y_val)

    # Repeat for test set
    X_test, y_test = [], []
    for i in range(timesteps, len(test_data)):
        X_test.append(test_data[i - timesteps:i, filtered_data.columns.get_indexer(feature_columns)])
        y_test.append(test_data[i, parameter_index])
    X_test, y_test = np.array(X_test), np.array(y_test)

    # Reshape input data
    X_train = np.reshape(X_train, (X_train.shape[0], timesteps * len(feature_columns)))
    X_val = np.reshape(X_val, (X_val.shape[0], timesteps * len(feature_columns)))
    X_test = np.reshape(X_test, (X_test.shape[0], timesteps * len(feature_columns)))

    # Define and compile neural network model
    model = Sequential()
    model.add(Dense(100, input_shape=(timesteps * len(feature_columns),), activation='relu'))
    model.add(Dense(50, activation='relu'))
    model.add(Dense(1))
    optimizer = Adam(learning_rate=0.001)
    model.compile(optimizer=optimizer, loss='mean_squared_error')

    # Train model
        model.fit(X_train, y_train, epochs=parameters[parameter]['epochs'], batch_size=256, validation_data=(X_val,
y_val))

    # Save model
    model.save(f'ann_{parameter}_{"_".join(features)}_model.h5')

    # Make predictions
    train_predictions = model.predict(X_train).flatten()
    val_predictions = model.predict(X_val).flatten()
    test_predictions = model.predict(X_test).flatten()

    # Calculate evaluation metrics
    train_r2 = r2_score(y_train, train_predictions)
    train_rmse = np.sqrt(mean_squared_error(y_train, train_predictions))
    train_mae = mean_absolute_error(y_train, train_predictions)

    val_r2 = r2_score(y_val, val_predictions)
    val_rmse = np.sqrt(mean_squared_error(y_val, val_predictions))
    val_mae = mean_absolute_error(y_val, val_predictions)

    test_r2 = r2_score(y_test, test_predictions)
    test_rmse = np.sqrt(mean_squared_error(y_test, test_predictions))
    test_mae = mean_absolute_error(y_test, test_predictions)
```

```
# Print evaluation metrics
print("Parameter:", parameter)
print("Features:", features)
print("Train R^2 Score:", train_r2)
print("Train RMSE:", train_rmse)
print("Train MAE:", train_mae)
print("Validation R^2 Score:", val_r2)
print("Validation RMSE:", val_rmse)
print("Validation MAE:", val_mae)
print("Test R^2 Score:", test_r2)
print("Test RMSE:", test_rmse)
print("Test MAE:", test_mae)
print()

# Plot actual vs. predicted values for training set
plt.figure(figsize=(8, 6))
plt.plot(range(100), y_train[:100], 'r-', linestyle='--', label='Actual (Training)')
plt.plot(range(100), train_predictions[:100], 'g-', label='Predicted (Training)')
plt.xlabel('Sample')
plt.ylabel(parameter + ' Concentration')
plt.title(f'Actual vs. Predicted {parameter} Concentration (Training)')
plt.legend()
plt.tight_layout()
plt.show()

# Plot actual vs. predicted values for validation set
plt.figure(figsize=(8, 6))
plt.plot(range(100), y_val[:100], 'r-', linestyle='--', label='Actual (Validation)')
plt.plot(range(100), val_predictions[:100], 'g-', label='Predicted (Validation)')
plt.xlabel('Sample')
plt.ylabel(parameter + ' Concentration')
plt.title(f'Actual vs. Predicted {parameter} Concentration (Validation)')
plt.legend()
plt.tight_layout()
plt.show()

# Plot actual vs. predicted values for test set
plt.figure(figsize=(8, 6))
plt.plot(range(100), y_test[:100], 'r-', linestyle='--', label='Actual (Testing)')
plt.plot(range(100), test_predictions[:100], 'g-', label='Predicted (Testing)')
plt.xlabel('Sample')
plt.ylabel(parameter + ' Concentration')
plt.title(f'Actual vs. Predicted {parameter} Concentration (Testing)')
plt.legend()
plt.tight_layout()
plt.show()
```

Appendix I provides a detailed overview of the data processing and model development steps employed to analyze particulate matter (PM) concentrations, including PM1.0, PM2.5, and PM10. Initially, the dataset undergoes a median filtering process to reduce noise and ensure the integrity of the PM readings. The data is then prepared for machine learning by encoding categorical variables and constructing features based on time series inputs. The analysis involves training artificial neural networks (ANNs) on various feature combinations to predict PM concentrations, using different models for each type of PM. The models are trained and validated using distinct data subsets, with performance metrics such as R-squared, RMSE, and MAE computed for the training, validation, and test datasets. The appendix further includes visual comparisons between the actual and predicted values for each PM type, showcasing the model's effectiveness. Each trained model is saved for future reference, highlighting the methodological rigor and attention to detail in capturing the complex relationships between environmental factors and PM levels.

# Appendix II: Implementation of SVR PM Model

```
import numpy as np
import pandas as pd
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
from sklearn.svm import SVR
import matplotlib.pyplot as plt
import joblib
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from scipy.signal import medfilt

# Load the data
data = pd.read_csv('PM_Training_DataSet_2024.csv')  # Replace 'PM_Training_DataSets.csv' with your dataset filename

# Convert non-numerical data in "Orientation" to numerical
le = LabelEncoder()
data['Orientation'] = le.fit_transform(data['Orientation'])

# Select the parameters to consider ('PM1.0', 'PM2.5', 'PM10')
parameters = ['PM1.0', 'PM2.5', 'PM10']

# Set SVR parameters
svr_params = {
    'C': 100,  # Penalty parameter C
    'epsilon': 3,  # Epsilon parameter in the epsilon-insensitive loss function
    'kernel': 'rbf'  # Kernel function (e.g., 'linear', 'rbf', 'poly')
}

# Define the number of time steps
timesteps = 5  # Number of previous time steps to consider

# Define feature combinations
feature_combinations = [
    ['Euclidean_D', 'Orientation'],
    ['Wind speed', 'Euclidean_D', 'Orientation'],
    ['Temperature', 'Euclidean_D', 'Orientation'],
    ['Humidity', 'Euclidean_D', 'Orientation'],
    ['Temperature', 'Humidity', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Temperature', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Humidity', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Temperature', 'Humidity', 'Euclidean_D', 'Orientation']
]

# Train and save models for each parameter and feature combination
for parameter in parameters:
    for features in feature_combinations:
        # Combine target parameter and selected features
        selected_columns = [parameter] + features

        # Apply 1D median signal filtering to selected variables
        for column in selected_columns:
            column_index = data.columns.get_loc(column)
            data[column] = medfilt(data[column], kernel_size=3)  # Set window size to 5

        # Save the filtered data
        filtered_filename = f'filtered_PM_Training_DataSets_{parameter}_{"_".join(features)}.csv'
        data[selected_columns].to_csv(filtered_filename, index=False)

        # Load the filtered data for training
        filtered_data = pd.read_csv(filtered_filename)

        # Split the data into training, validation, and testing sets
        train_size = int(len(filtered_data) * 0.6)
        validation_size = int(len(filtered_data) * 0.2)
```

```
train_data = filtered_data.iloc[:train_size, :].values
validation_data = filtered_data.iloc[train_size:train_size + validation_size, :].values
test_data = filtered_data.iloc[train_size + validation_size:, :].values

# Prepare the training data
X_train, y_train = [], []
for i in range(timesteps, len(train_data)):
    X_train.append(train_data[i - timesteps:i, :])
    y_train.append(train_data[i, :])
X_train, y_train = np.array(X_train), np.array(y_train)

# Prepare the validation data
X_validation, y_validation = [], []
for i in range(timesteps, len(validation_data)):
    X_validation.append(validation_data[i - timesteps:i, :])
    y_validation.append(validation_data[i, :])
X_validation, y_validation = np.array(X_validation), np.array(y_validation)

# Prepare the testing data
X_test, y_test = [], []
for i in range(timesteps, len(test_data)):
    X_test.append(test_data[i - timesteps:i, :])
    y_test.append(test_data[i, :])
X_test, y_test = np.array(X_test), np.array(y_test)

# Reshape the input data for SVR
X_train = np.reshape(X_train, (X_train.shape[0], timesteps * len(selected_columns)))
X_validation = np.reshape(X_validation, (X_validation.shape[0], timesteps * len(selected_columns)))
X_test = np.reshape(X_test, (X_test.shape[0], timesteps * len(selected_columns)))

# Build the SVR model
model = SVR(**svr_params)

# Train the model on the combined training and validation data
X_combined = np.concatenate((X_train, X_validation), axis=0)
y_combined = np.concatenate((y_train[:, 0], y_validation[:, 0]))  # Target parameter is at index 0

model.fit(X_combined, y_combined)

# Make predictions
train_predictions = model.predict(X_train)
validation_predictions = model.predict(X_validation)
test_predictions = model.predict(X_test)

# Compute evaluation metrics
train_r2 = r2_score(y_train[:, 0], train_predictions)
validation_r2 = r2_score(y_validation[:, 0], validation_predictions)
test_r2 = r2_score(y_test[:, 0], test_predictions)

train_rmse = np.sqrt(mean_squared_error(y_train[:, 0], train_predictions))
validation_rmse = np.sqrt(mean_squared_error(y_validation[:, 0], validation_predictions))
test_rmse = np.sqrt(mean_squared_error(y_test[:, 0], test_predictions))

train_mae = mean_absolute_error(y_train[:, 0], train_predictions)
validation_mae = mean_absolute_error(y_validation[:, 0], validation_predictions)
test_mae = mean_absolute_error(y_test[:, 0], test_predictions)

# Print the evaluation metrics for each parameter and feature combination
print(f"Parameter: {parameter}, Features: {features}")
print("Train R^2 Score:", train_r2)
print("Validation R^2 Score:", validation_r2)
print("Test R^2 Score:", test_r2)
print("Train RMSE:", train_rmse)
print("Validation RMSE:", validation_rmse)
print("Test RMSE:", test_rmse)
print("Train MAE:", train_mae)
```

```
print("Validation MAE:", validation_mae)
print("Test MAE:", test_mae)
print()

# Save the trained model
model_filename = f"svr_{parameter}_model_{('_').join(features)}.joblib"
joblib.dump(model, model_filename)

# Plot the graph (only first 100 samples)
plt.figure(figsize=(8, 6))
plt.plot(range(100), y_train[:100, 0], 'r--', label='Actual (Training)')
plt.plot(range(100), train_predictions[:100], 'g', label='Predicted (Training)')
plt.xlabel('Sample')
plt.ylabel(parameter + ' Concentration')
plt.title(f'Actual vs. Predicted {parameter} Concentration (Training)')
plt.legend()
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 6))
plt.plot(range(100), y_validation[:100, 0], 'r--', label='Actual (Validation)')
plt.plot(range(100), validation_predictions[:100], 'g', label='Predicted (Validation)')
plt.xlabel('Sample')
plt.ylabel(parameter + ' Concentration')
plt.title(f'Actual vs. Predicted {parameter} Concentration (Validation)')
plt.legend()
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 6))
plt.plot(range(100), y_test[:100, 0], 'r--', label='Actual (Testing)')
plt.plot(range(100), test_predictions[:100], 'g', label='Predicted (Testing)')
plt.xlabel('Sample')
plt.ylabel(parameter + ' Concentration')
plt.title(f'Actual vs. Predicted {parameter} Concentration (Testing)')
plt.legend()
plt.tight_layout()
plt.show()
```

The code applies Support Vector Regression (SVR) to predict PM1.0, PM2.5, and PM10 concentrations using various feature combinations such as Euclidean Distance, Orientation, Wind Speed, Temperature, and Humidity. The SVR model is trained using filtered time-series data, incorporating a 5-timestep window for each feature combination. The data is split into training, validation, and testing sets, and the model's performance is evaluated using $R^2$, RMSE, and MAE metrics. The code also saves the trained models and visualizes the predicted versus actual concentrations for the first 100 samples in the training, validation, and testing sets.

# Appendix III: Implementation of LSTM PM Model

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, LabelEncoder, OneHotEncoder
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import LSTM, Dense, Dropout
from keras.optimizers import Adam
import matplotlib.pyplot as plt

def median_filter_1d(signal, window_size=3):
    """Performs median filtering on a 1D signal."""
    filtered_signal = np.empty_like(signal)
    for i in range(signal.shape[0]):
        median = np.median(signal[max(0, i - window_size + 1):i + 1])
        filtered_signal[i] = median
    return filtered_signal

# Load the original dataset
data = pd.read_csv('PM_Training_DataSet_2024.csv')

# Apply median filter to PM1.0, PM2.5, and PM10 columns
columns_to_filter = ['PM1.0', 'PM2.5', 'PM10']
for column in columns_to_filter:
    data[column] = median_filter_1d(data[column])

# Save the filtered data to a new CSV file
filtered_data = data.copy()
filtered_data.to_csv('PM_Filtered_Training_DataSet_JAN.csv', index=False)

print("Filtered data saved as 'PM_Filtered_Training_DataSet_JAN.csv'.")

# Load and preprocess the filtered data
data = pd.read_csv('PM_Filtered_Training_DataSet_JAN.csv')

# Convert non-numeric values in "Orientation" to numerical
encoder = LabelEncoder()
data['Orientation'] = encoder.fit_transform(data['Orientation'])

# One-hot encode the "Orientation" column
onehot_encoder = OneHotEncoder(sparse=False)
orientation_encoded = onehot_encoder.fit_transform(data[['Orientation']])
data = pd.concat([data, pd.DataFrame(orientation_encoded)], axis=1)
data.drop('Orientation', axis=1, inplace=True)

# Define parameters and corresponding epoch values
parameters_epochs = {'PM1.0': 15, 'PM2.5': 20, 'PM10': 15}

# Define the number of time steps
timesteps = 3  # Number of previous time steps to consider

# Define filter size window
filter_window = 2

# Train and save models for each parameter and feature combination
features_combinations = [
    ['Euclidean_D', 'Orientation'],
    ['Wind speed', 'Euclidean_D', 'Orientation'],
    ['Temperature', 'Euclidean_D', 'Orientation'],
    ['Humidity', 'Euclidean_D', 'Orientation'],
    ['Temperature', 'Humidity', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Temperature', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Humidity', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Temperature', 'Humidity', 'Euclidean_D', 'Orientation']
```

```
]

for parameter in parameters_epochs.keys():
    for features in features_combinations:
        # Get the column index of the selected parameter
        parameter_index = data.columns.get_loc(parameter)

        # Select all features including the parameter column
        selected_features = features + [parameter]
        feature_indices = [i for i, col in enumerate(data.columns) if col in selected_features]

        # Split the data into training, validation, and testing sets
        train_size = int(len(data) * 0.8)
        validation_size = int(len(data) * 0.1)
        test_size = len(data) - train_size - validation_size

        train_data, validation_data, test_data = np.split(data, [train_size, train_size + validation_size])

        # Fit scaler only on the target parameter column
        parameter_scaler = MinMaxScaler(feature_range=(0, 1))
        parameter_scaler.fit(data.iloc[:, parameter_index].values.reshape(-1, 1))

        # Apply median filter to the target parameter column
        train_data[parameter] = median_filter_1d(train_data[parameter].values, window_size=filter_window)
        validation_data[parameter] = median_filter_1d(validation_data[parameter].values, window_size=filter_window)
        test_data[parameter] = median_filter_1d(test_data[parameter].values, window_size=filter_window)

        # Prepare the training data
        X_train, y_train = [], []
        for i in range(timesteps, len(train_data)):
            X_train.append(train_data.iloc[i - timesteps:i, feature_indices].values)
            y_train.append(train_data.iloc[i, parameter_index])
        X_train, y_train = np.array(X_train), np.array(y_train)

        # Prepare the validation data
        X_validation, y_validation = [], []
        for i in range(timesteps, len(validation_data)):
            X_validation.append(validation_data.iloc[i - timesteps:i, feature_indices].values)
            y_validation.append(validation_data.iloc[i, parameter_index])
        X_validation, y_validation = np.array(X_validation), np.array(y_validation)

        # Prepare the testing data
        X_test, y_test = [], []
        for i in range(timesteps, len(test_data)):
            X_test.append(test_data.iloc[i - timesteps:i, feature_indices].values)
            y_test.append(test_data.iloc[i, parameter_index])
        X_test, y_test = np.array(X_test), np.array(y_test)

        # Reshape the input data for LSTM [samples, time steps, features]
        X_train = np.reshape(X_train, (X_train.shape[0], timesteps, len(feature_indices)))
        X_validation = np.reshape(X_validation, (X_validation.shape[0], timesteps, len(feature_indices)))
        X_test = np.reshape(X_test, (X_test.shape[0], timesteps, len(feature_indices)))

        # Build the LSTM model with dropout
        model = Sequential()
        model.add(LSTM(50, input_shape=(timesteps, len(feature_indices)), return_sequences=True))
        model.add(Dropout(0.2))
        model.add(LSTM(50))
        model.add(Dropout(0.2))
        model.add(Dense(1))

        # Set the learning rate
        learning_rate = 0.01

        # Compile the model with a custom learning rate
        optimizer = Adam(learning_rate=learning_rate)
        model.compile(optimizer=optimizer, loss='mean_squared_error')
```

```
    # Train the model
                        model.fit(X_train, y_train, epochs=parameters_epochs[parameter], batch_size=256,
validation_data=(X_validation, y_validation))

    # Save the trained model with the parameter name and features
    model.save(f'lstm_{parameter}_{"_".join(features)}_model.h5')

    # Make predictions
    train_predictions = model.predict(X_train)
    validation_predictions = model.predict(X_validation)
    test_predictions = model.predict(X_test)

    # Reshape the predictions for plotting
    train_predictions = train_predictions.flatten()
    validation_predictions = validation_predictions.flatten()
    test_predictions = test_predictions.flatten()

    # Compute evaluation metrics
    train_r2 = r2_score(y_train, train_predictions)
    train_r2 = max(0, train_r2)  # Avoid negative R^2 scores
    train_rmse = np.sqrt(mean_squared_error(y_train, train_predictions))
    train_mae = mean_absolute_error(y_train, train_predictions)

    validation_r2 = r2_score(y_validation, validation_predictions)
    validation_r2 = max(0, validation_r2)
    validation_rmse = np.sqrt(mean_squared_error(y_validation, validation_predictions))
    validation_mae = mean_absolute_error(y_validation, validation_predictions)

    test_r2 = r2_score(y_test, test_predictions)
    test_r2 = max(0, test_r2)
    test_rmse = np.sqrt(mean_squared_error(y_test, test_predictions))
    test_mae = mean_absolute_error(y_test, test_predictions)

    # Print the evaluation metrics for each parameter and features
    print(f"Parameter: {parameter}, Features: {features}")
    print("Train R^2 Score:", train_r2)
    print("Train RMSE:", train_rmse)
    print("Train MAE:", train_mae)
    print("Validation R^2 Score:", validation_r2)
    print("Validation RMSE:", validation_rmse)
    print("Validation MAE:", validation_mae)
    print("Test R^2 Score:", test_r2)
    print("Test RMSE:", test_rmse)
    print("Test MAE:", test_mae)
    print()

    # Plot the graphs for Predicted Vs. Actual for Training, Validation, and Testing
    plt.figure(figsize=(8, 6))

    # Training plot
    plt.plot(range(100), y_train[:100], 'r-', linestyle='--', label='Actual')
    plt.plot(range(100), train_predictions[:100], 'g-', label='Predicted')
    plt.xlabel('Sample')
    plt.ylabel(parameter + ' Concentration')
    plt.title(f'Training Set - Actual vs. Predicted {parameter} Concentration - {", ".join(features)}')
    plt.legend()
    plt.tight_layout()
    plt.show()

    # Validation plot
    plt.figure(figsize=(8, 6))
    plt.plot(range(100), y_validation[:100], 'r-', linestyle='--', label='Actual')
    plt.plot(range(100), validation_predictions[:100], 'g-', label='Predicted')
    plt.xlabel('Sample')
    plt.ylabel(parameter + ' Concentration')
    plt.title(f'Validation Set - Actual vs. Predicted {parameter} Concentration - {", ".join(features)}')
```

```
plt.legend()
plt.tight_layout()
plt.show()

# Testing plot
plt.figure(figsize=(8, 6))
plt.plot(range(100), y_test[:100], 'r-', linestyle='--', label='Actual')
plt.plot(range(100), test_predictions[:100], 'g-', label='Predicted')
plt.xlabel('Sample')
plt.ylabel(parameter + ' Concentration')
plt.title(f'Testing Set - Actual vs. Predicted {parameter} Concentration - {", ".join(features)}')
plt.legend()
plt.tight_layout()
plt.show()

# New figure window for correlation plot
plt.figure(figsize=(8, 6))

# Define correlation line coordinates
correlation_line_x = np.linspace(0, 1, 100)
correlation_line_y = correlation_line_x

# Plot actual vs predicted (first 100 samples)
plt.scatter(y_test[:7500], test_predictions[:7500], color='g', label='Predicted')
plt.plot(y_test[:7500], y_test[:7500], color='r', marker='o', linestyle='-', label='Actual')
plt.plot(correlation_line_x, correlation_line_y, color='black', linestyle='--', label='Correlation Line')
plt.xlabel(f'Actual {parameter} Concentration')
plt.ylabel(f'Predicted {parameter} Concentration')
plt.title(f'Actual vs. Predicted {parameter} Concentration - Correlation - {", ".join(features)}')
plt.legend()
plt.tight_layout()

# Show the correlation plot
plt.show()
```

The code implements a Long Short-Term Memory (LSTM) model to predict PM1.0, PM2.5, and PM10 concentrations based on various feature combinations like Euclidean Distance, Orientation, Wind Speed, Temperature, and Humidity. The model is trained on median-filtered data to remove noise, with 3 previous timesteps used for input sequences. The LSTM model is built with two LSTM layers, each followed by a dropout layer, and trained using the Adam optimizer with a custom learning rate. The model's performance is evaluated using $R^2$, RMSE, and MAE metrics for training, validation, and testing datasets. The code also saves the trained models and visualizes actual vs. predicted concentrations, including a correlation plot to assess prediction accuracy.

## Appendix IV: Implementation of Random Forest Regression PM Model

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import joblib

# Load the dataset
df = pd.read_csv("PM_Training_DataSet_2024.csv")

# Perform 1D signal median filtering for PM1.0, PM2.5, and PM10
filter_size = 85  # Adjust the filter size based on your data and noise characteristics

def median_filter_1d(signal, filter_size):
    """Performs median filtering on a 1D signal."""
    padded_signal = np.pad(signal, ((filter_size - 1) // 2, (filter_size - 1) // 2), mode='constant')
    filtered_signal = np.empty_like(signal)
    for i in range(signal.shape[0]):
        window = padded_signal[i:(i + filter_size)]
        median = np.median(window)
        filtered_signal[i] = median
    return filtered_signal

# Apply median filtering to PM1.0, PM2.5, and PM10 columns
columns_to_filter = ['PM1.0', 'PM2.5', 'PM10']
for column in columns_to_filter:
    df[column] = median_filter_1d(df[column].values, filter_size)

# Save the filtered dataset
filtered_df_path = "Filtered_PM_Training_DataSets_4.csv"
df.to_csv(filtered_df_path, index=False)
print(f'Filtered dataset saved as '{filtered_df_path}'.")

# Load the filtered dataset
df = pd.read_csv(filtered_df_path)

# Convert non-numeric values in the Orientation column to numeric
df['Orientation'] = pd.Categorical(df['Orientation']).codes

# Function to compute and print evaluation metrics
def evaluate_model(model, X, y_true, label):
    y_pred = model.predict(X)
    r2 = max(0, r2_score(y_true, y_pred))  # Ensure R^2 score is non-negative
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    mae = mean_absolute_error(y_true, y_pred)

    accuracy = 100 - mae

    print(f"\nEvaluation for {label}:")
    print(f"R^2 Score: {r2:.6f}")
    print(f"Root Mean Squared Error (RMSE): {rmse:.6f}")
    print(f"Mean Absolute Error (MAE): {mae:.6f}")

# Define different feature combinations
feature_combinations = [
```

```
    ['Euclidean_D', 'Orientation'],
    ['Wind speed', 'Euclidean_D', 'Orientation'],
    ['Temperature', 'Euclidean_D', 'Orientation'],
    ['Humidity', 'Euclidean_D', 'Orientation'],
    ['Temperature', 'Humidity', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Temperature', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Humidity', 'Euclidean_D', 'Orientation'],
    ['Wind speed', 'Temperature', 'Humidity', 'Euclidean_D', 'Orientation']
]

# Define target columns
target_columns = ['PM1.0', 'PM2.5', 'PM10']

# Train and save models for different feature combinations
for features in feature_combinations:
    for target_column in target_columns:
        # Extract features and targets from the dataset
        X = df[features].values
        y = df[target_column].values

        # Scale the features using MinMaxScaler
        scaler = MinMaxScaler()
        X_scaled = scaler.fit_transform(X)

        # Split the data into training, validation, and testing sets
        X_train, X_temp, y_train, y_temp = train_test_split(X_scaled, y, test_size=0.4, random_state=42)
            X_validation, X_test, y_validation, y_test = train_test_split(X_temp, y_temp, test_size=0.5,
random_state=42)

        # Set adjusted parameters for RandomForestRegressor
        params = {
            'n_estimators': 500,  # Increase number of trees
            'max_depth': None,  # Allow deeper trees
            'min_samples_split': 50,  # Require fewer samples to split
            'min_samples_leaf': 1,
            'random_state': 42
        }

        # Train a Random Forest Regressor
        model = RandomForestRegressor(**params)
        model.fit(X_train, y_train)

        # Save the trained model
        model_filename = f'{target_column}_model_{"_".join(features)}.joblib'
        joblib.dump(model, model_filename)

        # Evaluate the model
        evaluate_model(model, X_train, y_train, f"{target_column} ({', '.join(features)}) (Training)")
            evaluate_model(model, X_validation, y_validation, f"{target_column} ({', '.join(features)})
(Validation)")
        evaluate_model(model, X_test, y_test, f"{target_column} ({', '.join(features)}) (Testing)")

        print(f"Trained model for {target_column} with features {features} saved as '{model_filename}'.")

'''
        # Plot Actual vs. Predicted for the first 100 samples
        plt.figure(figsize=(10, 5))
        plt.plot(y[:100], 'r--', label='Actual')
        plt.plot(model.predict(X_scaled[:100]), 'g-', label='Predicted')
        plt.title(f"Actual vs. Predicted for {target_column} ({', '.join(features)})")
```

```
    plt.xlabel("Sample")
    plt.ylabel(target_column)
    plt.legend()
    plt.show()
'''
```

This appendix provides the code implementation and methodology used to develop Random Forest regression models for predicting particulate matter concentrations (PM1.0, PM2.5, and PM10) based on various environmental and geometric features. The script includes steps for data preprocessing, feature scaling, model training, and evaluation using key metrics such as $R^2$ score, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). The process involved the application of median filtering for noise reduction, followed by training the Random Forest models on different combinations of features, and saving the trained models for future use. The performance of each model was evaluated across training, validation, and testing datasets to ensure robustness and accuracy.

# Appendix V: Publication

IOP Publishing Environ. Res. Commun. 7 (2025) 085020 https://doi.org/10.1088/2515-7620/adfb46

Environmental Research Communications

PAPER

# Development of integrated machine learning model for estimation of spatial distribution of particulate matter pollutant in air

**Alex Mwololo Kimuya**[1], **Dickson Mwenda Kinyua**[1,2] **and Daniel Maitethia Memeu**[1,*]

[1] Department of Physical Sciences, Meru University of Science and Technology, Kenya
[2] Department of Pure and Applied Sciences, Kirinyaga University, Kenya
* Author to whom any correspondence should be addressed.

**E-mail:** dmaitethia@must.ac.ke

**Keywords:** air pollution, particulate matter (PM), air quality sensors, ANN (artificial neural network), LSTM (long short-term memory), random forest, SVR (support vector regression)

Supplementary material for this article is available online

OPEN ACCESS

RECEIVED
21 May 2025

REVISED
4 August 2025

ACCEPTED FOR PUBLICATION
13 August 2025

PUBLISHED
28 August 2025



## Abstract

Accurate estimation of particulate matter (PM) concentration in air is crucial for understanding and mitigating air pollution. Conventional PM measurement systems often provide only single-point, instantaneous readings, limiting their ability to create comprehensive spatial pollution maps. This study aims to address this limitation by leveraging Machine Learning (ML) techniques to predict PM concentration using data from a limited number of sensor nodes. ML training spatiotemporal dataset was collected over a month, encompassing PM1.0, PM2.5, and PM10 concentrations, weather parameters, and spatial information. Four machine learning models namely; Long Short-Term Memory (LSTM), Artificial Neural Network (ANN), Support Vector Regression (SVR), and Random Forest (RF), were evaluated for their ability to predict the spatial distribution of PM concentration at three different sites. The analysis revealed that ANN consistently outperformed other models across different feature combinations. Within a 160-meter radius of a central sensor node at (First Site), the ANN model achieved an average prediction accuracy above 86.0% for PM concentrations, with RMSE and MAE values of 1.72 $\mu g\ m^{-3}$ and 0.80 $\mu g\ m^{-3}$, respectively. The inclusion of weather parameters and feature engineering improved model performance by 12% to 18%, compared to models using only geometric features. When applied to (Second Site), the ANN model maintained an accuracy of 85%, demonstrating strong intra-environment generalizability. However, performance declined at the (Third Site), located 3 kilometers away in a peri-urban market setting, where microclimatic and topographic variability resulted in a lower prediction accuracy of 59.2%. This study demonstrates the effectiveness of a machine learning-based approach to overcome the limitations of single-point PM sensors and predict PM concentration across a region. The results highlight the superior performance of ANN and emphasize the importance of incorporating weather parameters and feature engineering in model training. These findings are promising and point to a new approach of developing a practical model for spatial estimation of PM distribution from measurements obtained from sensor networks.

## Appendix VI: Plagiarism Report

DrillBit

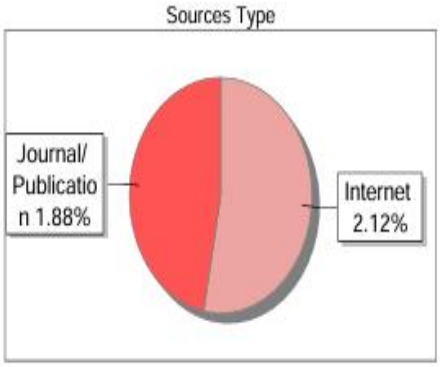
The Report is Generated by DrillBit Plagiarism Detection Software

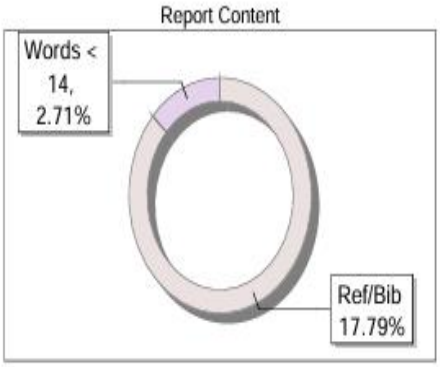
*Submission Information*

| | |
|---|---|
| Author Name | ALEX MWOLOLO KIMUYA |
| Title | DEVELOPMENT OF MACHINE LEARNING MODEL FOR ESTIMATION OF SPATIAL DISTRIBUTION OF PARTICULATE MATTER POLLUTANT IN AIR |
| Paper/Submission ID | 2179521 |
| Submitted by | mmusungu@must.ac.ke |
| Submission Date | 2024-07-29 09:56:38 |
| Total Pages, Total Words | 123, 25097 |
| Document type | Thesis |

*Result Information*

Similarity 4 %




*Exclude Information*

| | |
|---|---|
| Quotes | Not Excluded |
| References/Bibliography | Excluded |
| Source: Excluded < 14 Words | Not Excluded |
| Excluded Source | 0 % |
| Excluded Phrases | Not Excluded |

*Database Selection*

| | |
|---|---|
| Language | English |
| Student Papers | Yes |
| Journals & publishers | Yes |
| Internet or Web | Yes |
| Institution Repository | Yes |

A Unique QR Code use to View/Download/Share Pdf File