PREDICTIVE ANALYSIS OF DUST LEVELS IN VIT

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Electronics and communication

by

Umar Khan (21BEC0100)

Rohan Kumar (21BEC0185)

Aryan Mohapatra(21BEC0721)

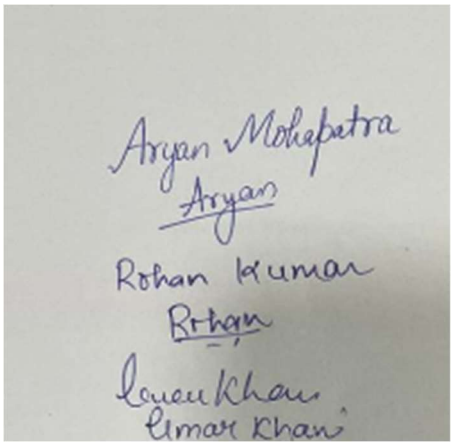
Under the guidance of

Dr. Abhishek Naraya Tripathi
Department of Electronics and Communications Engineering

November 2024

DECLARATION

We hereby declare that the thesis entitled “PREDICTIVE ANALYSIS OF DUST LEVELS IN VIT” submitted by Aryan Mohapatra (21BEC0721), Rohan Kumar (21BEC0185) and Umar Khan (21BEC0100), for the completion of the course “BECE497J – Project 1"to the school of electronics engineering, Vellore Institute of Technology, Vellore is Bonafide work carried out by me under the supervision of D r . Abhishek Narayan Tripathi. I further declare that the work reported in this thesis has not been submitted previously to this institute or anywhere.



Place : Vellore

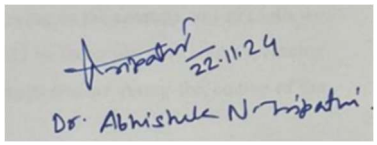
Date : 15/11/24

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled “PREDICTIVE ANALYSIS OF DUST LEVELS IN VIT” submitted by Umar Khan (21BEC0100), Rohan Kumar (21BEC0185) and Aryan Mohapatra (21BEC0721) SENSE, VIT, for the completion of the course “BECE497J – Project 1”, is a Bonafide work carried out by him under my supervision during the period, 01.08.2024 to 15.11.2024 as per the VIT code of academic and research ethics.

We further declare that the work reported in this thesis has not been submitted previously to this institute or anywhere.



Place : Vellore

Date :15/11/24

Signature of the Guide

Internal Examiner

ACKNOWLEDGEMENTS

We wish to express our heartfelt gratitude to **Dr. G.Viswanathan**, Chancellor, VIT University, Vellore for providing facilities for the seventh semester project.

We are highly grateful to our Vice Presidents, **Shri. Sankar Viswanathan, Shri. Sekar Viswanathan and Shri. G.V. Selvam**, Vice Chancellor **Dr. V. S. Kanchana Bhaskaran**, Pro- Vice Chancellor **Dr. Partha Sharathi Mallick** for providing necessary resources.

We would like to express our special gratitude and thanks to our internal guide **Dr. Abhishek Narayan Tripathi**, Assistant Professor Sr. Grade 1, School of Electronics Engineering whose esteemed guidance and immense support encouraged us to complete the project successfully. We thank the Management of VIT University for permitting us to use the library resources. We also thank all the faculty members of VIT University for giving us the courage and strength we needed to complete our goals. This acknowledgement would be incomplete without expressing our wholehearted thanks to our family and friends who motivated us during the course of the work.

Umar Khan

Rohan Kumar

Aryan Mohapatra

Executive Summary

The suggested method of applying machine learning algorithms for predictive analysis of dust levels offers a substantial advancement over conventional techniques for monitoring and forecasting dust levels. Traditional approaches sometimes rely on manual monitoring, which is frequently expensive and time-consuming and may not give consumers real-time warnings about excessive dust levels in the area.

In contrast, real-time monitoring and prediction of dust levels may be done more cheaply and effectively by using machine learning techniques. The data may be pre-processed, separated into training and testing datasets, and utilized to create a model that can forecast future dust levels by using data collected using a dust sensor and stored in Excel sheets. This method may be used to notify users in real-time when there are excessive amounts of dust in the air, allowing them to take preventative action to lower the risk of respiratory issues and other health risks brought on by high dust levels.

Using machine learning techniques to anticipate dust levels has the potential to completely change the way dust monitoring is done. This strategy can enable more frequent and thorough monitoring of dust levels in locations with high dust concentrations, such as construction sites, mines, and factories, by offering a cost-effective and efficient method of monitoring and forecasting dust levels. This can aid in
lowering the likelihood of respiratory issues and other health risks brought on by excessive environmental dust levels.

Furthermore, this method's ramifications go beyond only monitoring and forecasting dust. This method can open the door for the development of comparable applications in other sectors including air pollution monitoring, water quality monitoring, and climate prediction by showcasing the potential of machine learning algorithms for predictive analysis of environmental parameters. Overall, the proposed method of using machine learning algorithms for predictive analysis of dust levels represents a significant advancement in dust monitoring techniques, with implications for the health and safety of people working in high-dust environments as well as for the creation of related applications in other fields.

CONTENTS *Page No.*

List of Figures

List of Tables

List of Abbreviations

GMT	Gauss-Markov Theorem
ML	Machine Learning
PM	Particulate Matter
PMS	Particulate Measuring System

Symbols and Notation

y	Dependent Variable
x	Independent Variable
=	Equals
k	Number of features (or predictors)
N	Number of Trees
TTL	Transistor-Transistor Logic

1. INTRODUCTION

1.1 Literature Review

The increasing particulate matter (PM) pollution due to industrial activities, vehicular emissions, and urban development has prompted extensive research on air quality monitoring and predictive models. PM10 and PM2.5 particles are particularly concerning as they pose significant health risks, especially respiratory and cardiovascular diseases, due to their ability to penetrate deep into the lungs. Consequently, machine learning has become instrumental in predicting PM levels, allowing for real-time alerts and preventive actions.

Several studies have implemented Polynomial Regression and Random Forest Regression models for predicting dust levels. Polynomial Regression is beneficial for modeling non-linear relationships, often providing a better fit for PM data that displays fluctuating patterns over time. Random Forest Regression, an ensemble method, combines predictions from multiple decision trees to improve accuracy. This model is particularly effective for handling non-linear interactions and has been applied in various industrial and urban contexts for air quality forecasting.

In addition to regression models, clustering techniques like K-means are increasingly employed to group PM data based on concentration patterns, which helps in identifying high-risk zones or periods. For instance, K-means clustering can categorize days or locations with similar pollution profiles, facilitating targeted intervention strategies.

However, existing approaches face limitations, such as sensitivity to outliers in polynomial models and the need for extensive data to ensure the accuracy of ensemble methods like Random Forest. Furthermore, short data collection periods can restrict a model's ability to generalize across diverse conditions and geographic areas. Despite these challenges, the integration of IoT sensors with machine learning models is a promising avenue for real-time dust monitoring, offering substantial improvements over manual air quality assessments by enabling timely warnings and contributing to sustainable urban and industrial practices.

1.2 Research Gap

The IoT-based dust level prediction project at VIT highlights gaps in existing dust monitoring methods. Primarily, the study's limited data collection period of 30 days restricts its effectiveness in capturing seasonal trends or long-term patterns in dust levels across various environments. Moreover, the prediction model relies on index-based inputs, which complicates specific date-based forecasts, limiting the model's practical utility. The model's accuracy maxes out at 86%, likely due to insufficient data, underscoring the need for larger, diverse datasets for robust predictions.

Additionally, the study only focuses on a localized area within the VIT Vellore campus, which limits the generalizability of the model to other areas with different environmental conditions. The machine learning algorithms employed also face challenges in classifying dust levels into actionable categories (e.g., high, medium, low), reducing the model's practicality for public health recommendations, such as mask usage advisories.

Addressing these gaps could involve extending data collection, improving spatial data diversity, and refining the model to provide categorical predictions. Such enhancements would make the model more versatile and applicable across broader environmental and health-focused applications. .

1.3 Problem Statement

Dust pollution, especially particulate matter (PM) such as PM1.0, PM2.5, and PM10, poses serious health and environmental challenges. These tiny particles, often generated from industrial processes, construction activities, vehicular emissions, and natural sources, have become pervasive in urban areas and industrial zones. The health implications of high dust levels are well-documented, as PM particles can penetrate the respiratory system and exacerbate conditions like asthma, bronchitis, and even cardiovascular diseases. For individuals in high-risk environments, such as construction sites, factories, or areas with high vehicle emissions, prolonged exposure to elevated dust levels can lead to severe health consequences.

Traditional air quality monitoring methods are limited, relying on static measurements and lacking the capability for real-time or predictive insights. Consequently, affected individuals are often unable to take preventive measures in time. There is a clear need for a system that can continuously monitor and predict dust levels, providing real-time alerts and forecasts to enable proactive responses. A data-driven approach that incorporates machine learning algorithms can meet this need by leveraging historical and real-time data to predict dust concentration levels with high accuracy.

This project aims to develop a predictive model using Polynomial Regression and Random Forest Regression to accurately forecast dust levels for different PM sizes. Additionally, K-means clustering will be employed to identify recurring dust concentration patterns, allowing for targeted responses in high-risk areas. Data collected from IoT-enabled sensors will be processed and analyzed to support real-time monitoring and prediction. By providing early warnings and actionable insights, the proposed system seeks to mitigate health risks associated with dust exposure and contribute to better urban air quality management practices. This project not only addresses immediate health concerns but also aligns with broader sustainability goals by promoting healthier, more resilient communities in the face of increasing dust pollution.

1.3.1 Relevance of the problem statement w.r.t to SDG

The problem of dust pollution and its health impacts directly aligns with several Sustainable Development Goals (SDGs) established by the United Nations, aimed at creating a healthier and more sustainable future. By developing a predictive system for dust level monitoring and alerts, this project contributes to the following SDGs:

SDG 3 - Good Health and Well-being:
Dust pollution, especially particulate matter like PM2.5 and PM10, poses significant health risks, leading to respiratory and cardiovascular diseases. This project aligns with SDG 3 by providing a real-time monitoring system that predicts and warns about high dust levels, helping to protect vulnerable populations and improve overall public health. Early alerts can reduce exposure to harmful particulates, promoting a healthier environment and preventing related diseases.

SDG 11 - Sustainable Cities and Communities:
Rapid urbanization and industrial activities contribute to increased dust pollution in cities. This project aids SDG 11 by enabling better air quality management in urban areas through continuous dust monitoring and predictive insights. With the ability to identify high-risk areas, the project supports city planners and policymakers in implementing dust mitigation measures, making cities more resilient and safer for residents.

SDG 13 - Climate Action:
Dust particles, especially from industries, construction, and vehicular emissions, contribute to air pollution and exacerbate climate change. By promoting dust monitoring and data-driven pollution management, this project supports SDG 13. The model provides actionable insights that help reduce dust pollution, contributing to cleaner air and minimizing the environmental impact associated with poor air quality.

SDG 9 - Industry, Innovation, and Infrastructure:
This project encourages the use of IoT and machine learning technologies to monitor environmental conditions in real time. By fostering innovation in air quality monitoring, it contributes to SDG 9, promoting the development of sustainable infrastructure and industrial practices that are mindful of their environmental footprint.
In summary, this project aligns with the SDGs by prioritizing health and sustainability through technology-driven air quality management solutions, supporting long-term goals for sustainable and resilient communities

2. PROJECT OBJECTIVE

The primary objective of this project is to develop an efficient and accurate predictive model for dust level monitoring, aimed at reducing the health risks associated with particulate matter (PM1.0, PM2.5, and PM10) exposure in high-dust environments. Leveraging machine learning algorithms and IoT-enabled sensors, the project seeks to provide real-time and predictive insights into dust concentrations to facilitate timely preventive actions.

Key Objectives:

Develop a Predictive Model:
Utilize machine learning algorithms, specifically Polynomial Regression and Random Forest Regression, to accurately forecast dust levels based on historical and real-time sensor data. These models are chosen to handle non-linear patterns in dust data and improve prediction accuracy across different PM levels.

Implement Clustering Analysis:
Apply K-means clustering to identify patterns and group high-risk dust levels by time and location, enabling more targeted monitoring and resource allocation for dust mitigation in specific areas.

Enable Real-time Monitoring and Alerts:
Integrate IoT-enabled sensors to continuously monitor dust levels and relay real-time data. The system will trigger alerts when dust levels exceed safe thresholds, providing early warnings to reduce exposure.

Enhance Data Processing and Visualization:
Develop a data pipeline for efficient cleaning, processing, and visualization of dust data to ensure the system provides clear and actionable insights for users and decision-makers.

Support Sustainable Urban Planning and Industrial Safety:
By delivering accurate predictions and identifying pollution hotspots, the project aims to aid urban planners, policymakers, and industries in implementing proactive measures to control dust pollution, aligning with broader health and sustainability goals.

3. PROPOSED WORK

3.1 Design Approach / System model / Algorithm

The proposed system for predictive analysis of dust levels integrates IoT-enabled sensors with machine learning algorithms to deliver real-time dust level monitoring and forecasting. The approach involves data collection, preprocessing, model training, and real-time prediction to provide actionable insights for mitigating dust pollution risks. Below is an outline of the system's design approach, model components, and algorithms used.

System Model

1. IoT Sensor Network:

   - PMS7003 Dust Sensor: An IoT-enabled dust sensor that continuously collects real-time data on particulate matter (PM1.0, PM2.5, PM10) concentrations.
   - Data Transfer via TTL Serial Cable: The sensor transmits data to a microcontroller (e.g., Arduino) that processes and forwards it to the data storage system for analysis.

2. Data Preprocessing Pipeline:

   - Data Cleaning: Removing outliers and handling missing values to ensure data quality. Outliers are identified and filtered based on the mean and standard deviation of each parameter.
   - Feature Engineering: Extract relevant features, such as PM concentration and time stamps, and index data by date and time for better analysis.

3. Machine Learning Model:

   - Polynomial Regression: Chosen for its capability to handle non-linear data, modeling the complex variations in dust levels over time. This regression model provides high accuracy for PM levels (PM1.0, PM2.5, and PM10).
   - Random Forest Regression:
     Used as an ensemble learning method, combining multiple decision trees to improve prediction accuracy and robustness. Random Forest is especially useful for capturing dust level trends with less overfitting risk.
   - K-means Clustering:
     Clusters dust concentration data to identify high-risk periods and patterns. This algorithm categorizes the data into clusters based on similarity, highlighting periods of elevated pollution levels and helping target specific times or areas for preventive measures.

4. Real-time Prediction and Alert System:
   - The trained model processes new sensor data in real-time, predicting dust levels and triggering alerts if PM concentrations exceed safe thresholds. The alert system is designed to notify stakeholders (e.g., workers, residents) to take preventive action during high dust exposure times.

5. Data Visualization and Reporting:
   - Visualization tools display real-time and historical data trends, including scatterplots, regression curves, and clustering heatmaps. This enables users to easily interpret data and make informed decisions based on dust level patterns.

Algorithm Summary

1. Polynomial Regression Algorithm:
   - Fits a polynomial equation to model the relationship between time (or index) and PM levels, handling non-linear patterns effectively.
   - Degree of the polynomial is optimized to balance model complexity and accuracy.

2. Random Forest Regression Algorithm:
   - An ensemble method that builds multiple decision trees using random samples of data and averages their predictions to improve accuracy.
   - Reduces the impact of overfitting by relying on multiple trees, which increases generalization.

3. K-means Clustering Algorithm:
   - Divides data into clusters by minimizing the distance between data points and their respective centroids.
   - Identifies high-risk periods by grouping time frames or locations with similar PM concentrations.

3.2 Technical Descriptions

1. **Hardware Components**

- **PMS7003 Dust Sensor**:
  - The PMS7003 is a digital particle concentration sensor that measures the concentration of particulate matter (PM1.0, PM2.5, PM10) in the air. It operates based on the laser scattering principle, where a laser beam scatters off particles, and the sensor calculates particle concentration in real time.
  - The sensor outputs particle data through a digital interface compatible with Arduino microcontrollers, which transmits data to a computer via a TTL serial download cable.

  - **Arduino Microcontroller**:
  - An Arduino microcontroller interfaces with the PMS7003 sensor, collecting data and sending it to a connected computer. The Arduino IDE is used for programming the microcontroller to handle data collection and transmission.

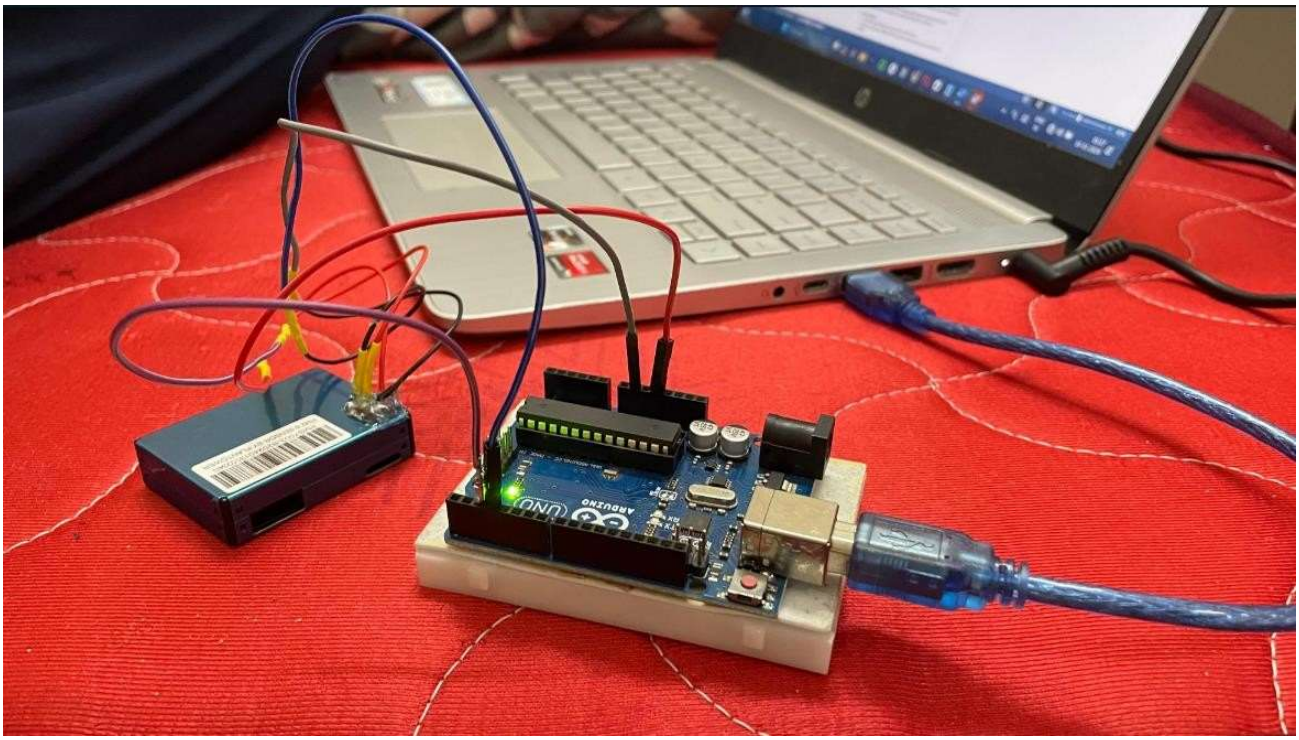


Figure 1. Hardware Setup

2. Software and Data Processing Tools

- **Data Collection and Storage**:
  - Data from the PMS7003 sensor is converted from the serial monitor to CSV format using "ArduSpreadsheet" for easier processing and analysis.

  - **Data Preprocessing**:
  - Outlier Removal: Outliers are identified and removed based on the range of mean ± 3 standard deviations.
  - Feature Scaling: Each data feature (PM1.0, PM2.5, PM10, and various particle sizes) is standardized to ensure all features contribute equally to the model.

Indexing: Each observation is indexed by date to support time-series analysis and prediction.

3. Machine Learning Algorithms

- **Polynomial Regression**:
  - Polynomial Regression models the non-linear relationship between particulate levels and time. By using a 4th-degree polynomial, the model captures fluctuations in dust concentration with high accuracy for smaller particles (PM1.0, PM2.5).
  - The model is trained and validated using metrics like Mean Squared Error (MSE) and R^2 to assess its prediction accuracy.

  - **Random Forest Regression**:
    - Random Forest, an ensemble method, constructs multiple decision trees, each on a random subset of data points and features. This approach reduces variance, improving the model's robustness against overfitting and noise in the data.
    - The algorithm combines individual tree predictions by averaging, providing a stable and accurate prediction for PM levels across different particle sizes.

4. Clustering Technique

- **K-means Clustering**:
  - K-means clustering groups data points based on particle concentrations, identifying high-risk periods or areas with elevated dust levels. The algorithm iteratively assigns each data point to one of kkk centroids, where kkk represents the predefined number of clusters.
  - After convergence, clusters represent patterns in the data, which are visualized to reveal dust concentration trends over time.

5. Evaluation Metrics

- **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)** quantify the model's prediction error.
- **R^2 Score** measures the proportion of variance explained by the model, indicating the model's overall fit.

4 HARDWARE/SOFTWARE TOOLS USED

Hardware Components

1. PMS7003 Dust Sensor:
   - Measures particulate matter (PM) concentrations (PM1.0, PM2.5, PM10).
   - Based on laser scattering, it detects suspended particles and calculates concentration, outputting data through a digital interface.
   - Provides real-time response with minimal error, making it suitable for continuous dust level monitoring.

2. Arduino Microcontroller:
   - Acts as an interface between the PMS7003 sensor and the computer.
   - Collects data from the sensor and transmits it via a TTL serial cable.
   - Programmed using Arduino IDE, enabling data handling, and integration with other components for effective communication.

3. TTL Serial Download Cable:
   - Connects the Arduino microcontroller to a computer for data transfer.
   - Converts serial data from the sensor to USB-compatible signals, ensuring smooth data logging and analysis.

Software Tools

1. Arduino IDE:
   - Used for programming the Arduino microcontroller to interact with the PMS7003 sensor.
   - Allows data collection, real-time monitoring, and configuration of data output for processing.

2. Jupyter Notebook:
   - Provides an environment for analyzing data collected from the sensor.
   - Used for data cleaning, feature engineering, and model training, with an emphasis on ease of visualization and code management.

3. Python Libraries:
   - Pandas and NumPy: Handle data manipulation and preprocessing, including removing outliers, scaling features, and organizing data for machine learning.
   - scikit-learn: Contains machine learning models, such as Polynomial Regression, Random Forest Regression, and K-means clustering, which are crucial for dust level prediction and clustering analysis.
   - Matplotlib and Seaborn: Used for visualizing data trends, regression curves, and cluster distributions, enabling clear insights into dust level variations.

4. ArduSpreadsheet:
   - Converts data from Arduino's serial monitor to CSV format.
   - Allows easy storage and handling of sensor data for further processing and analysis in Python.

5. RESULT ANANYSIS

System Architecture

The system architecture for the dust level prediction project is structured to enable seamless data collection, processing, model training, and real-time prediction. This architecture integrates IoT sensors with machine learning models and visualization tools, providing real-time insights into particulate matter levels.

1. Data Collection Layer:
   - PMS7003 Dust Sensor collects real-time data on particulate matter (PM1.0, PM2.5, PM10) concentrations.
   - Arduino Microcontroller retrieves data from the sensor and transmits it via a TTL serial download cable to a connected computer.

2. Data Preprocessing Layer:
   - Data is transferred from Arduino's serial output and converted into CSV format using ArduSpreadsheet.
   - Preprocessing includes cleaning, handling outliers, and scaling features using Pandas and NumPy in Jupyter Notebook.
   - Preprocessed data is indexed by date and organized for time-series analysis.

3. Machine Learning Model Layer:
   - Polynomial Regression: Used to model non-linear trends in PM levels and capture fluctuations effectively.
   - Random Forest Regression: Applies ensemble learning for robust predictions, improving the model's generalization and reducing the risk of overfitting.
   - K-means Clustering: Groups similar dust concentration levels to identify high-risk zones and periods, enhancing the system's ability to highlight critical dust exposure patterns.

4. Real-time Prediction and Alert System:
   - The trained models are deployed to provide real-time predictions based on incoming sensor data.
   - The system generates alerts when PM levels exceed safe thresholds, providing early warnings to help reduce health risks.

5. Visualization and Reporting Layer:
   - Visualization tools like Matplotlib and Seaborn display dust level trends, cluster distributions, and regression curves.
   - Real-time data is displayed as scatterplots and heatmaps, enabling easy identification of patterns and high-risk periods.

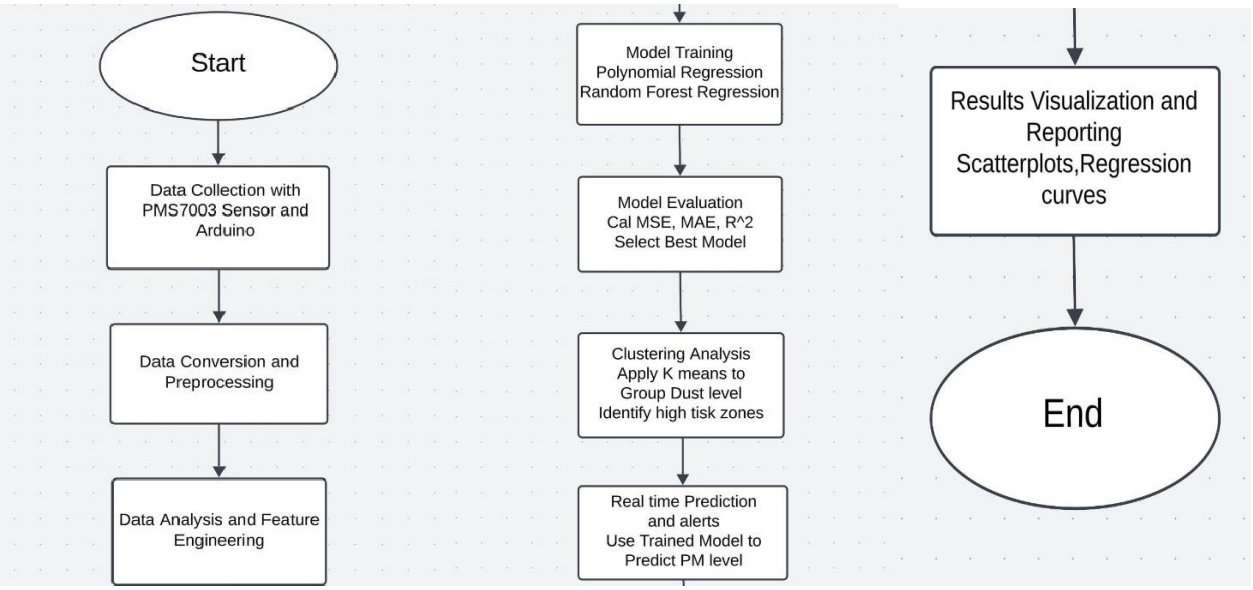


Figure 2. Architecture Flowchart

Result Analysis

1. Model Performance:
   - Polynomial Regression: Achieved accuracy scores of around 85% for PM1.0, PM2.5, and PM10 predictions, effectively capturing the non-linear trends in dust levels.
   - Random Forest Regression: Demonstrated high performance with lower error rates, offering a robust approach to predict PM levels across various particle sizes.
   - K-means Clustering: Identified significant dust concentration patterns by grouping similar data points, enabling targeted insights into dust exposure hotspots.

2. Evaluation Metrics:
   - Mean Squared Error (MSE) and Mean Absolute Error (MAE) were used to evaluate prediction accuracy. Both metrics indicated satisfactory model performance, with MSE highlighting areas with larger deviations.
   - R^2 Score: Scores near 0.85 indicate that the models explain a substantial portion of the variability in dust levels, making them reliable for predictive analysis.

3. Clustering Insights:
   - K-means clustering highlighted specific times and areas with recurring high dust levels, offering actionable insights for preventive measures. Visualization of clusters enabled the identification of dust concentration trends and facilitated real-time monitoring.

4. Visualization Outcomes:
   - Scatter plots, regression curves, and clustering heatmaps provided a clear visual representation of PM level trends and high-risk clusters.

Real-time predictions and clustering results helped inform users about dust level fluctuations, enabling timely actions to reduce exposure risks.

Polynomial Regression: -

```
[17]: df1=df.copy()
      df1['Data Index']=np.arange(len(df1.index))
      df1.head()
```

[17]:

	Types Of Partical	Concentration of particle	Unit	Date	Data Index
0	Warm4	7 (u	g/m3)	02.10.2024	0
1	PM10	:58	(ug/m3)	02.10.2024	1
5	PM2.5	:48	(ug/m3)	02.10.2024	2
6	PM10	:60	(ug/m3)	02.10.2024	3
9	Dust	Conc	entration:	02.10.2024	4

File Edit View Run Kernel Settings Help — Trusted

Code — JupyterLab — Python 3 (ipykernel)

```
[11]: import pandas as pd
      import numpy as np
      df = pd.read_csv(r"C:\Users\rohan\OneDrive\Desktop\test2\data.csv")
      df=df.dropna()
      df.head()
```

[11]:

	Types Of Partical	Concentration of particle	Unit	Date
0	Warm4	7 (u	g/m3)	02.10.2024
1	PM10	:58	(ug/m3)	02.10.2024
5	PM2.5	:48	(ug/m3)	02.10.2024
6	PM10	:60	(ug/m3)	02.10.2024
9	Dust	Conc	entration:	02.10.2024

```
[19]: df=df.dropna()
      df.tail()
```

[19]:

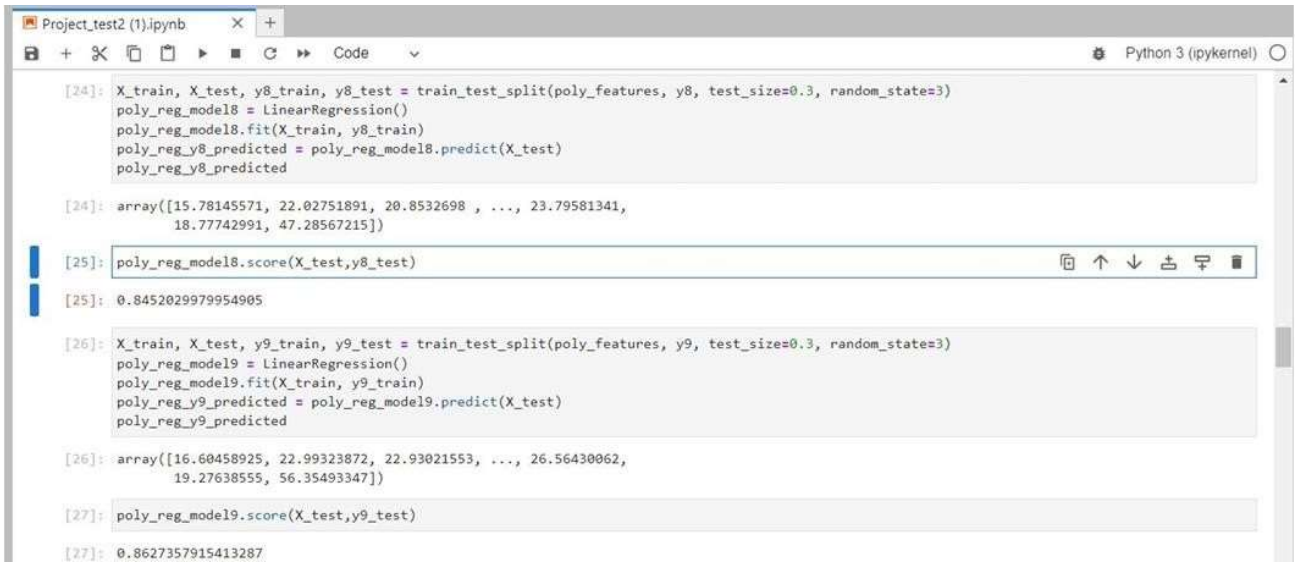
	Types Of Partical	Concentration of particle	Unit	Date
282	PM10	:64	(ug/m3)	02.10.2024
285	Dust	Conc	entration:	02.10.2024
286	PM1.0	:32	(ug/m3)	02.10.2024
287	PM2.5	:50	(ug/m3)	02.10.2024
288	PM10	:64	(ug/m3)	02.10.2024

In the above, it is seen that rows with NaN values are dropped. This can be verified by noticing the internally provided index values to each row of the dataframe before and after dropping them.

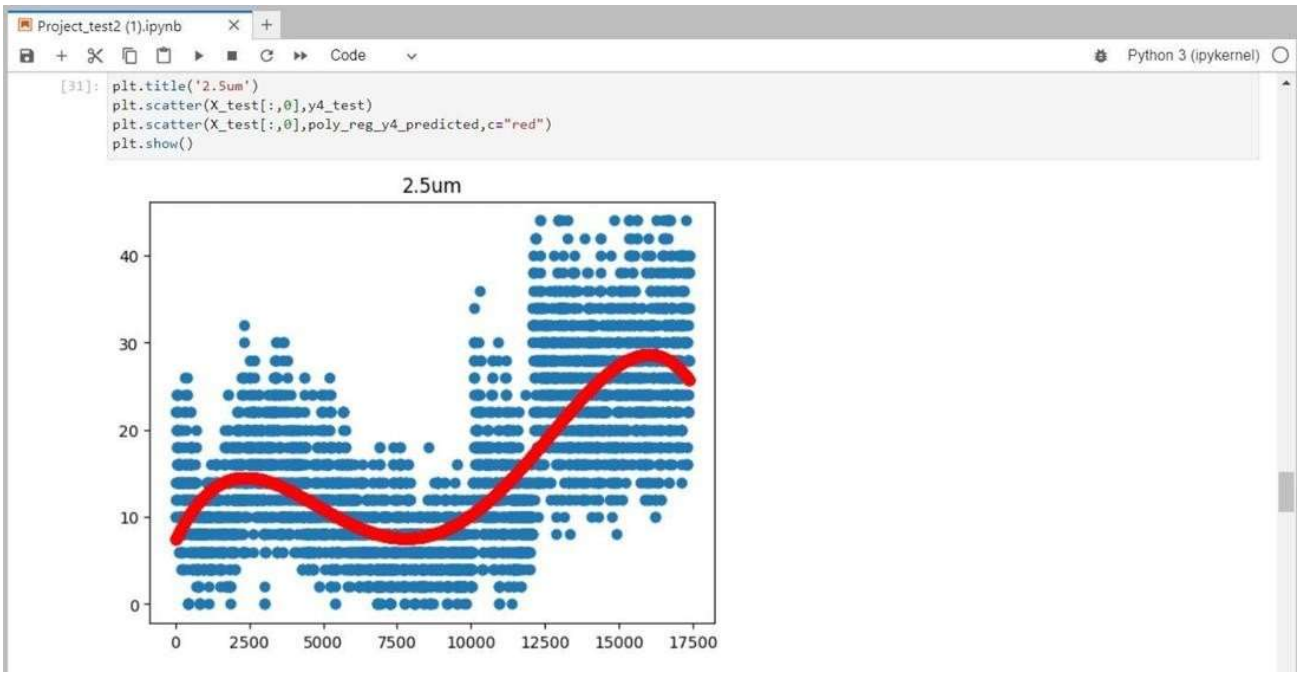
The above screenshots depict the training and testing of the 4th degree polynomial regression model for the '0.3um' variable. As seen, the model possesses an accuracy of around 87.92%.

The accuracy of the model for the '0.5um' variable is around 87.79% while that of the model for the '1um' variable is 84.92%

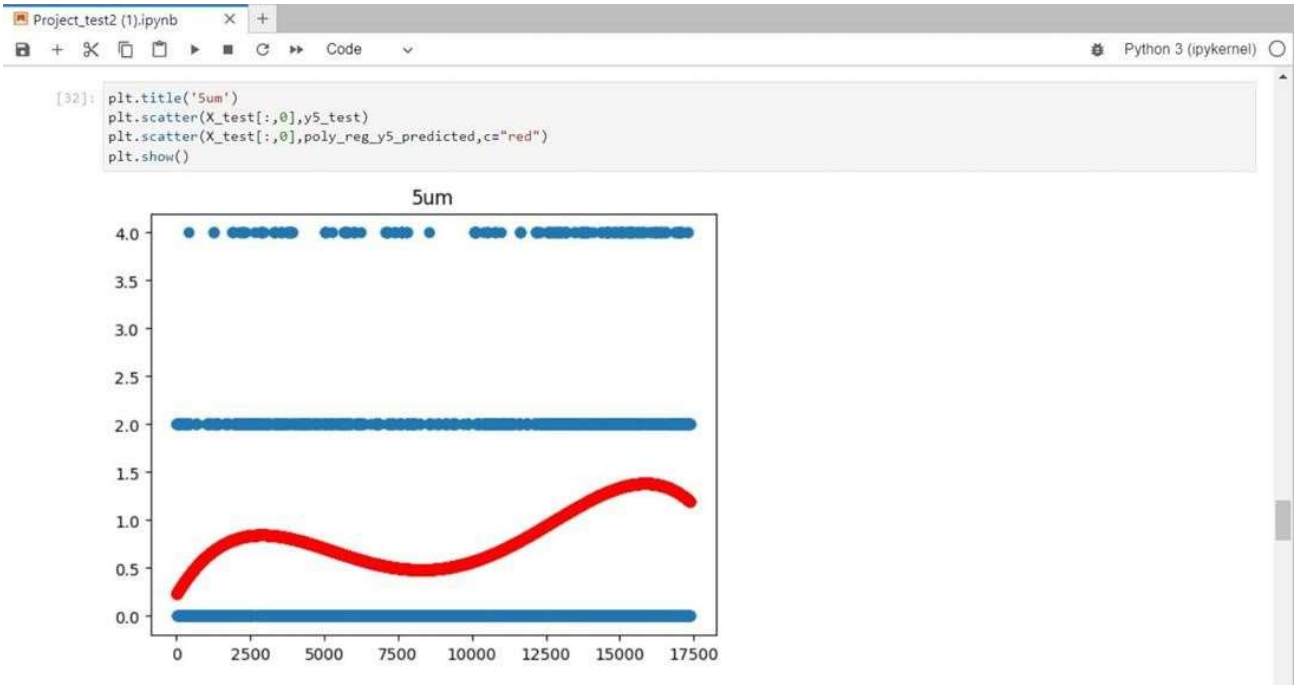
The accuracy of the model for the '2.5um' variable is around 54.65% while that of the model for the '5um' variable is just 5.43%. The low accuracies obtained for these are explained when the model and the scatterplots for the variables are shown further on.



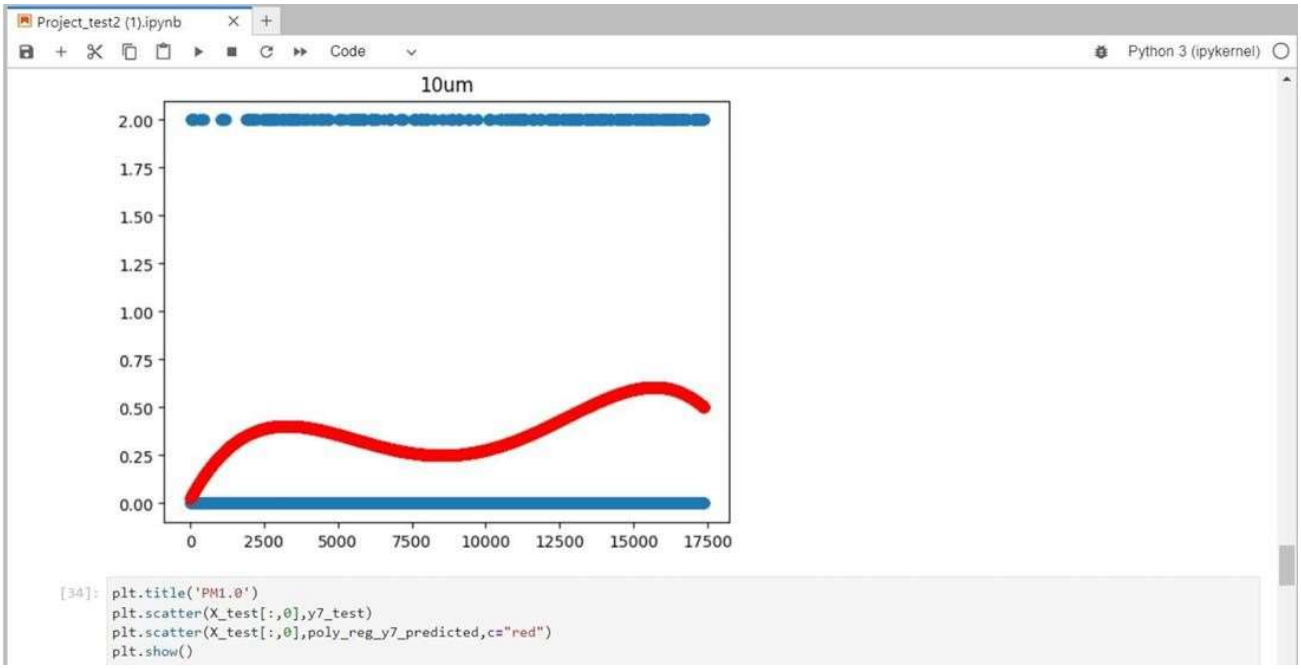
The accuracy of the model for the 'PM2.5' variable is around 84.52% while the accuracy for the 'PM10' variable is 86.27%.



Given above is the scatterplot and regression curve for the '2.5um' variable. From the scatterplot, it is observed that there are far fewer distinct values at a much lower scale than those of the previous variables. Thus, a deviation from the regression curve has much more impact on the accuracy of the model as compared to before, thus leading to the lower accuracy of 54.65%.



From the above, even without the accuracy score, it is clear that the model for the '5um' variable is very inaccurate. This can be explained by the fact that there are exactly 3 distinct values that the variable actually takes and at a scale very small deviations produce very high error. The regression curve, being continuous, cannot produce just 3 individual values and as such, cannot be accurate.

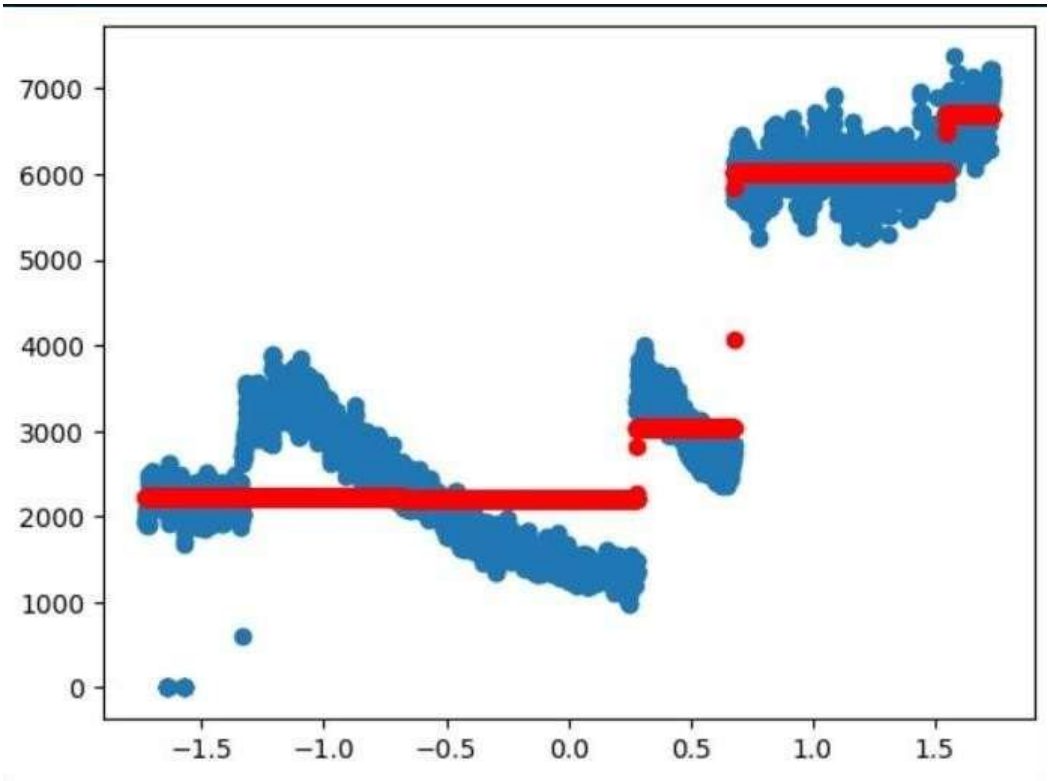


The 10um parameter takes just two values. As such, the regression model is wildly inaccurate for the same reason that the 5um model is.

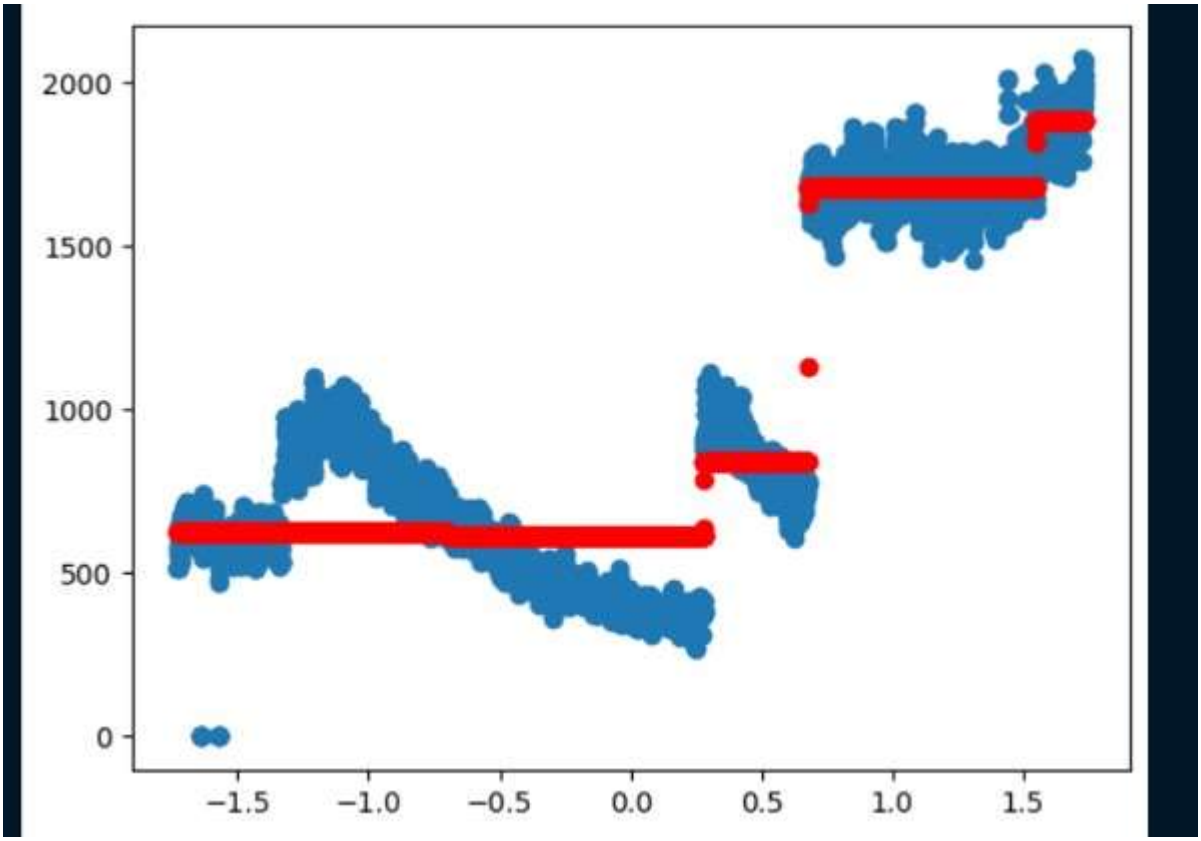
Limitations of using Polynomial Regression:

• The presence of one or two outliers in the data can seriously affect the results of the nonlinear analysis.
• There are fewer model validation tools for the detection of outliers in nonlinear regression than there are for linear regression.
• The fitted model is less reliable when it is built on a small sample size.
• Polynomial models have poor interpolator properties. High degree polynomials are notorious for oscillations between exact-fit values.
• Polynomial models have poor extrapolatory properties. Polynomials may provide good fits within the range of data, but they will frequently deteriorate rapidly outside the range of the data. • Polynomial models have poor asymptotic properties. By their nature, polynomials have a finite response for finite x values and have an infinite response if and only if the x value is infinite. Thus, polynomials may not model asymptotic phenomena very well.
• Polynomial models have a shape/degree trade-off. To model data with a complicated structure, the degree of the model must be high, indicating and the associated number of parameters to be estimated will also be high. This can result in highly unstable models.

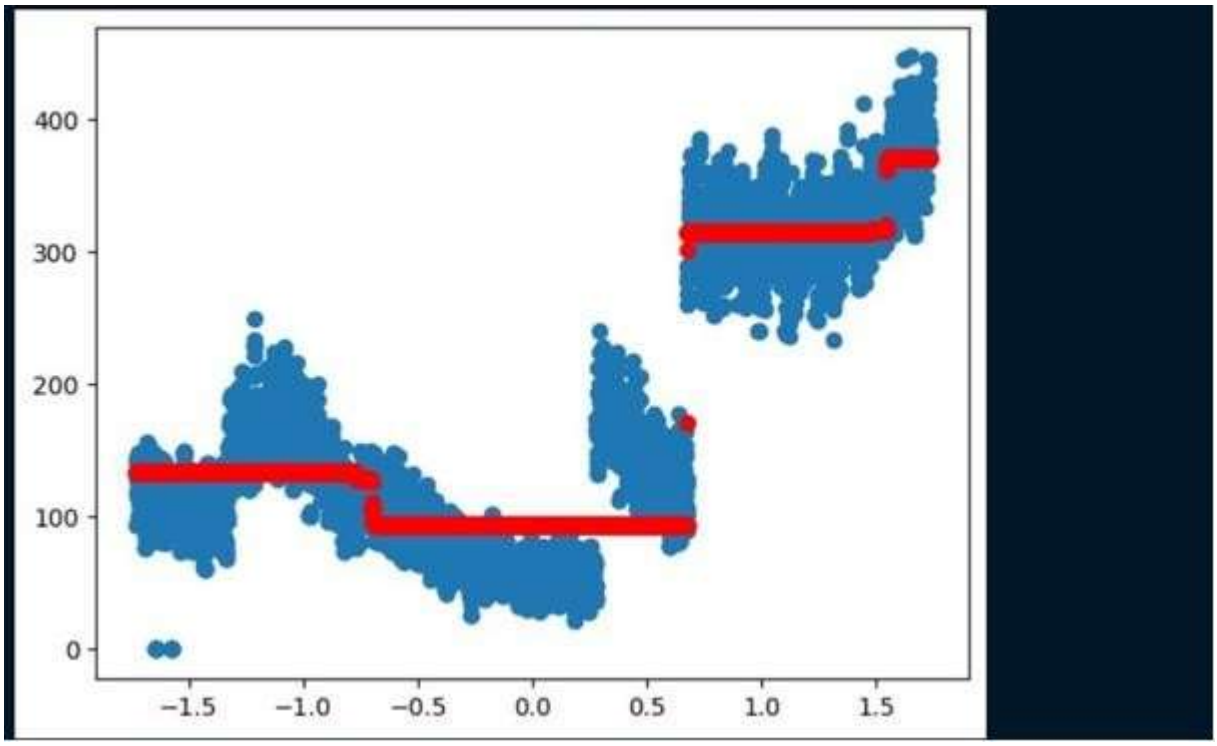
Random Forest Regression: -



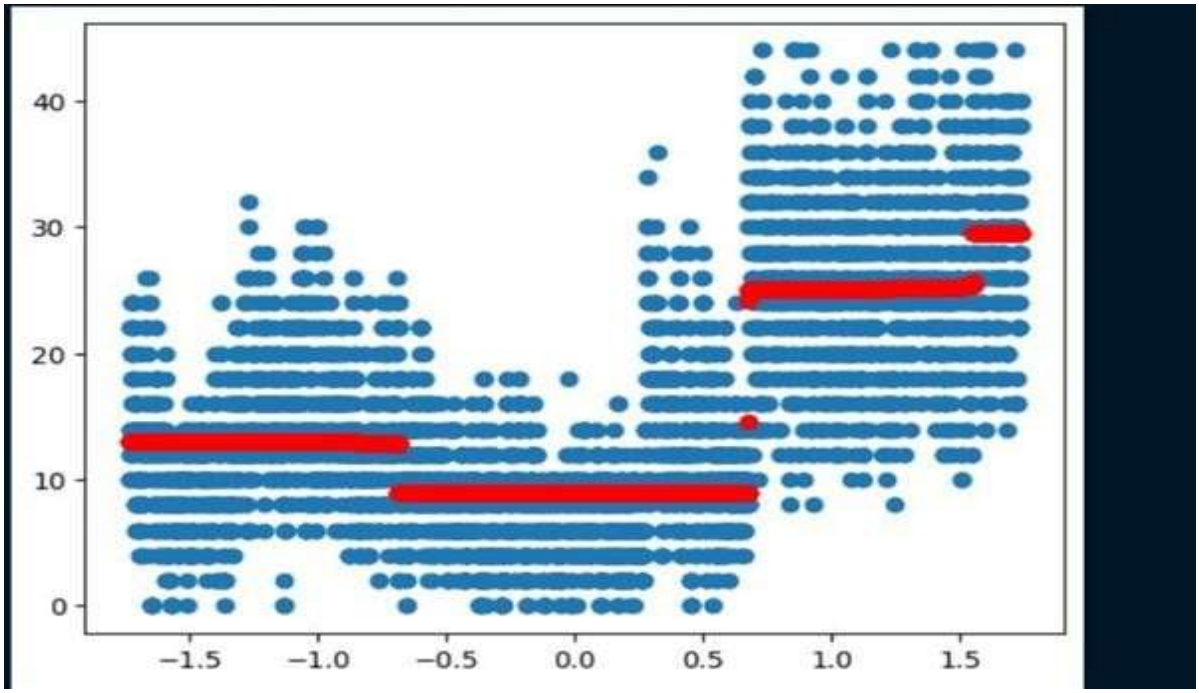
- Scatter plot showing the actual (blue points) vs. predicted (red points) values.
  The predicted values closely follow the distribution of actual values, indicating a good fit with an accuracy of 87.92%.
- The model effectively captures the trends and fluctuations for **0.3µm particles**.
  Minor deviations occur, but the predictions are generally consistent with the actual values.



- The scatter plot shows a strong correlation between actual and predicted values. Predicted values align closely with the actual data points. With an accuracy of 87.79%.
- The model effectively captures the trends and fluctuations for **0.5µm particles**. Minor deviations occur, but the predictions are generally consistent with the actual values



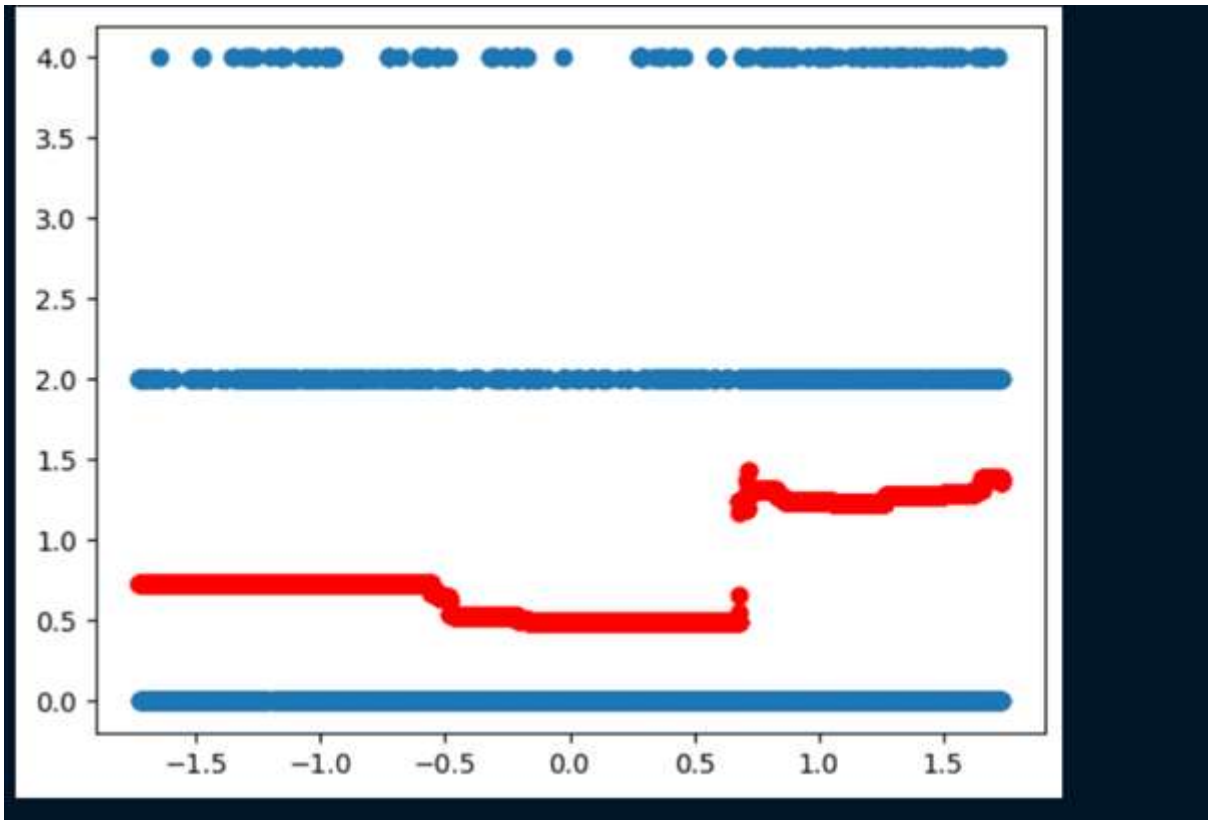
- A similar pattern of alignment between actual and predicted values, with slight deviations in some regions of **1.0µm Particle Concentrations**.
- With an accuracy of 84.92%
- Predictions are accurate for most data points, though some outliers result in a slightly lower accuracy compared to smaller particles.



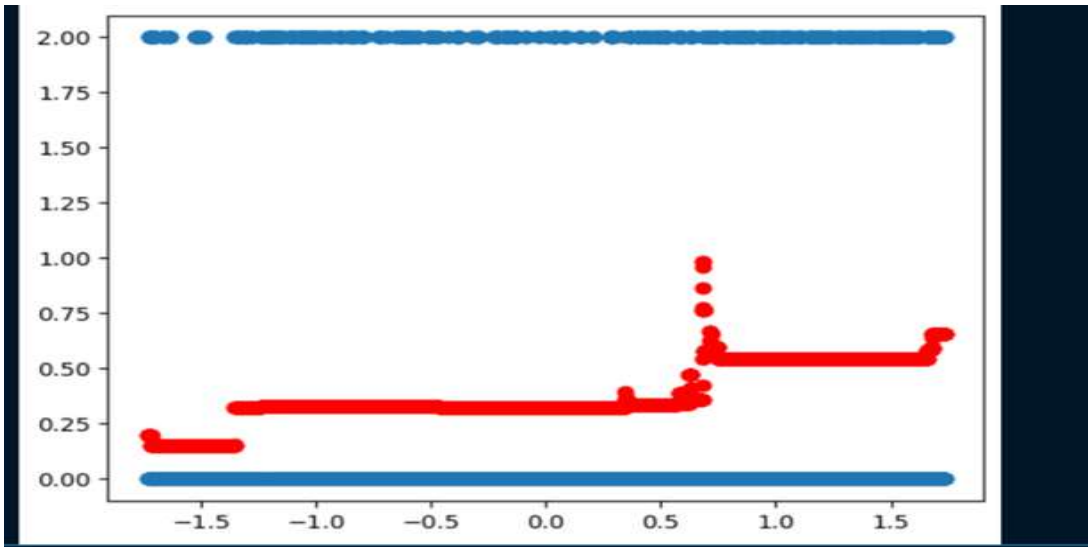
- Increased deviation between actual and predicted values compared to smaller

particles **2.5µm Particle Concentrations**.

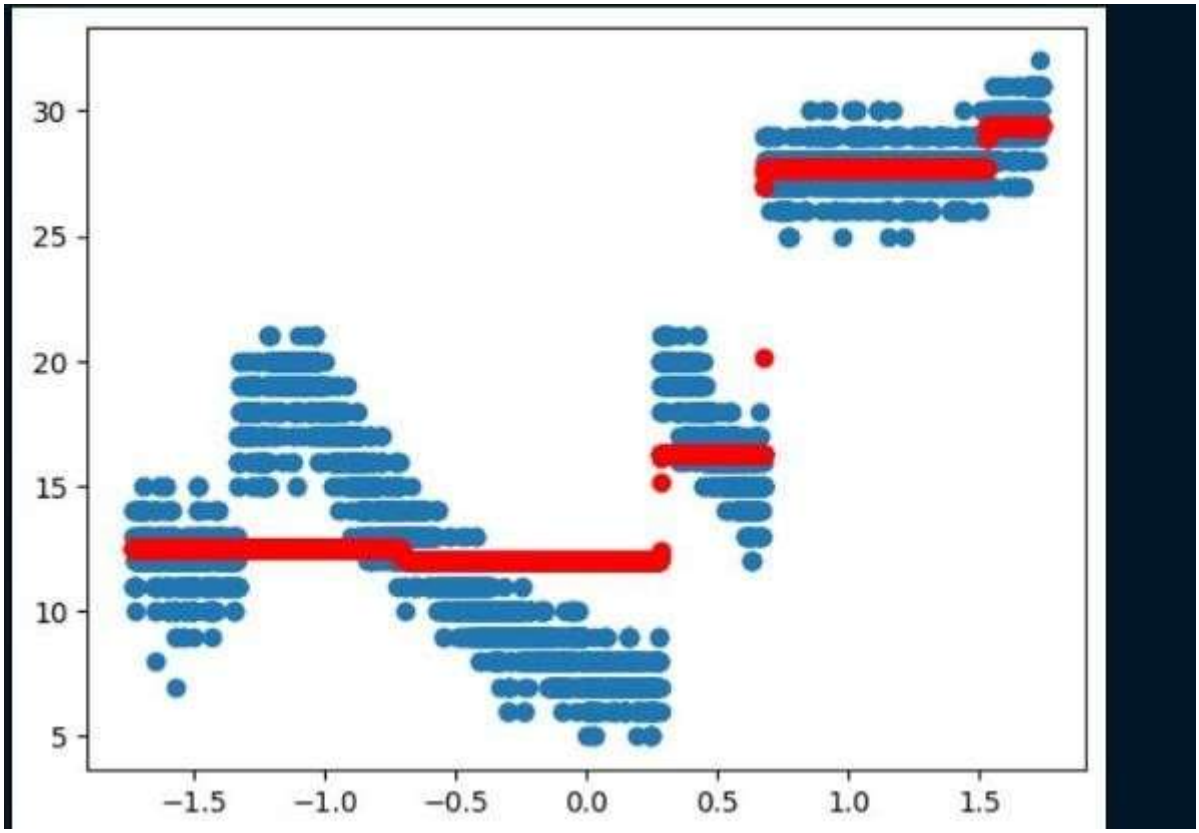
- With an accuracy of 54.65%
- The model struggles to fit certain patterns, resulting in less alignment.



- Significant scatter observed in the plot, indicating poor prediction accuracy. Predicted values do not follow the actual data distribution closely.
- With an accuracy of 5.43%
- Poor model performance is due to limited and discrete data for **5.0µm particles**. The lack of variability in the dataset makes it challenging for the model to learn meaningful patterns.

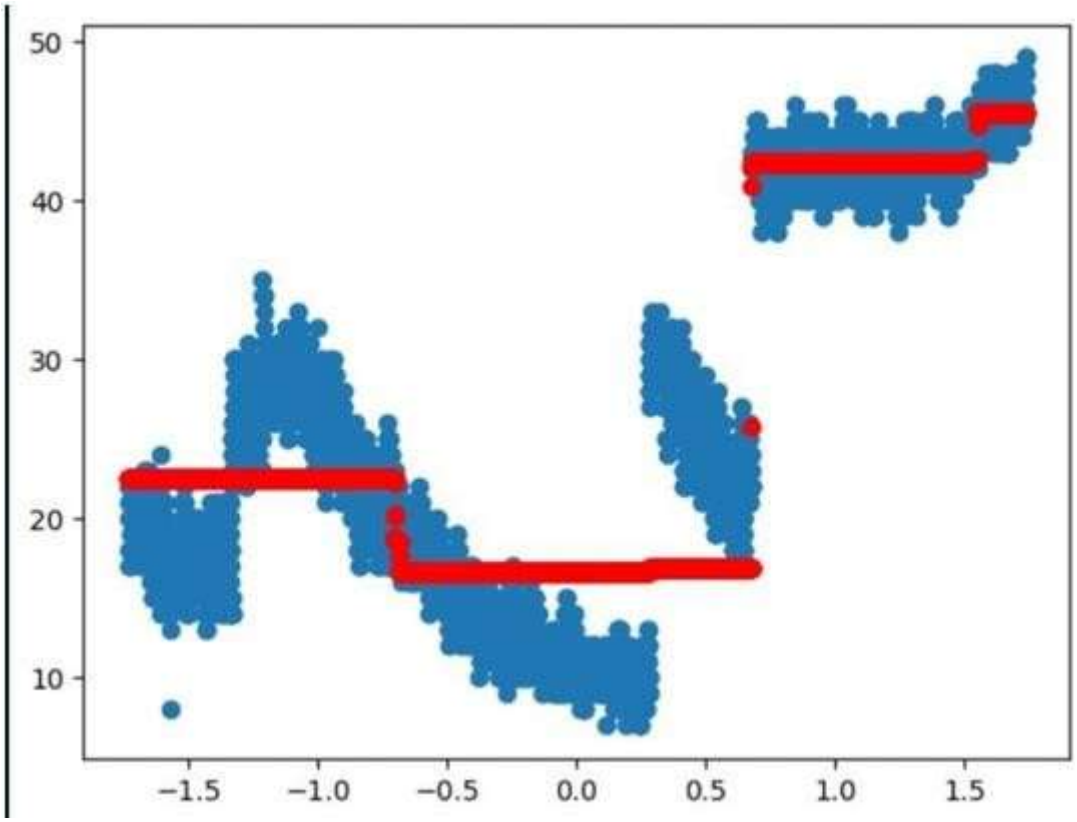


- The scatter plot shows extremely poor alignment between actual and predicted values of **10.0µm Particle Concentrations.**
- Most predicted values deviate significantly from the actual data.
- With an accuracy of 2.5%
- Like 5.0µm, the limited range of data points causes poor model performance.
- The model fails to generalize for this particle size due to insufficient data variability.

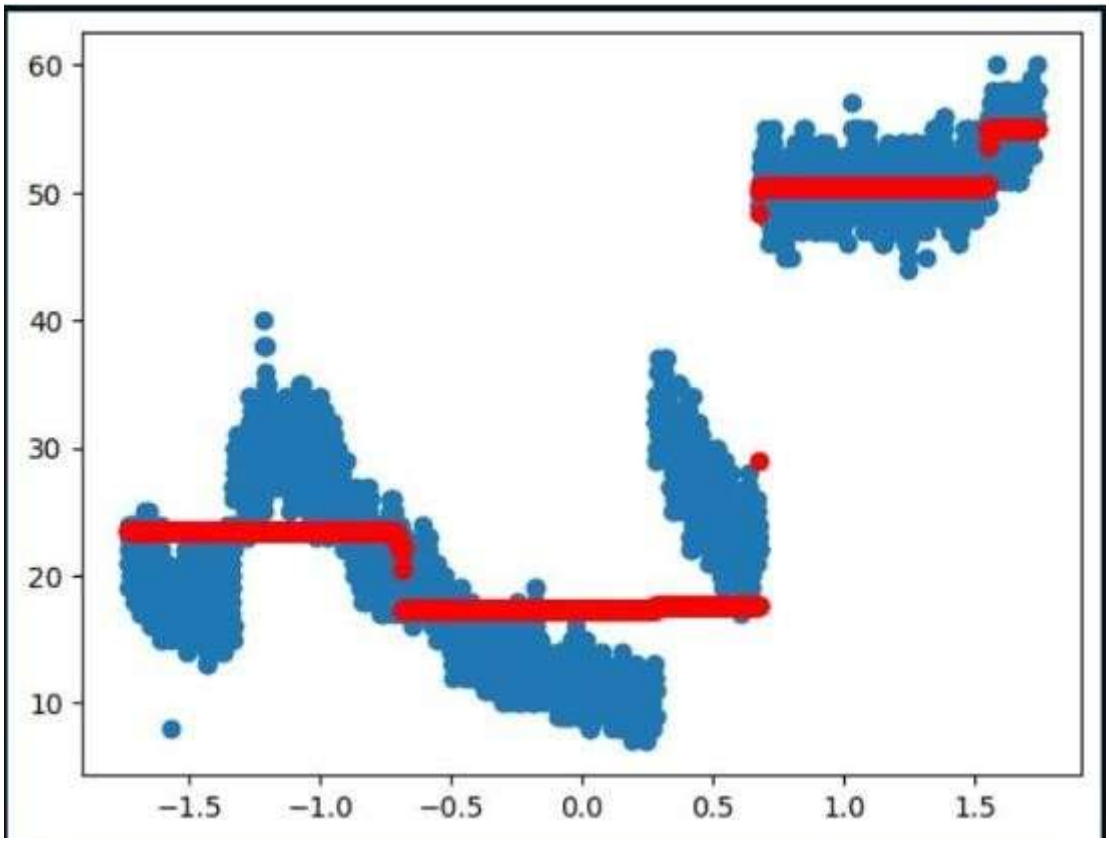


- A clear correlation between actual and predicted values, with minor deviations **PM1.0 Concentration.**

- With an accuracy of 84.97%
- The model accurately predicts PM1.0 levels, demonstrating its reliability for smaller particulate matter sizes.



- The scatter plot shows good alignment between actual and predicted values **PM2.5 Concentrations.**
- With an accuracy of 84.52%
- The model handles PM2.5 data effectively, producing predictions that closely match actual values.



- Predictions closely follow the trend of actual values, with only minor deviations
  PM10 Concentrations
- With an accuracy of 86.27%
- PM10 predictions are among the most accurate, indicating the model's strength in predicting this parameter.

Limitations of Random Forest Regression:

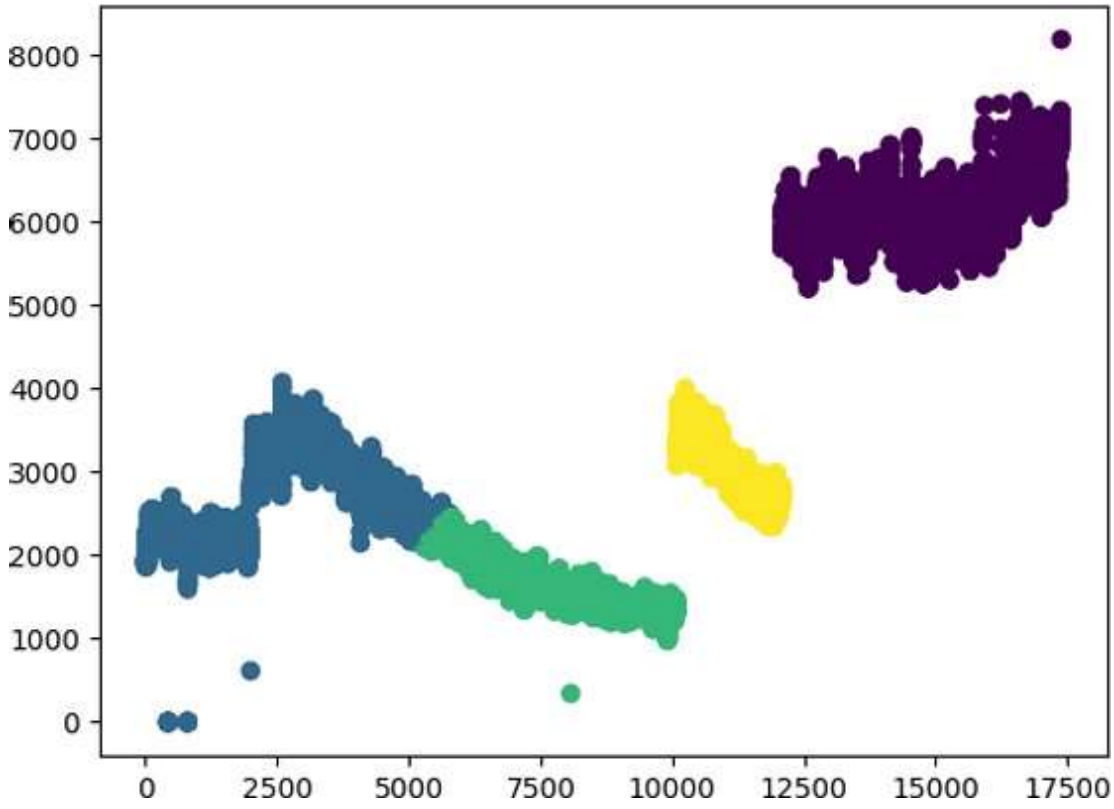
• Many trees are required to produce accurate results.
• Using more trees slows down the model.
• Overfitting may occur if too many trees are used.
• Does not perform well if data is unbalanced (seen in 5 and 10um data)
• Not suitable for predictions outside the data range.

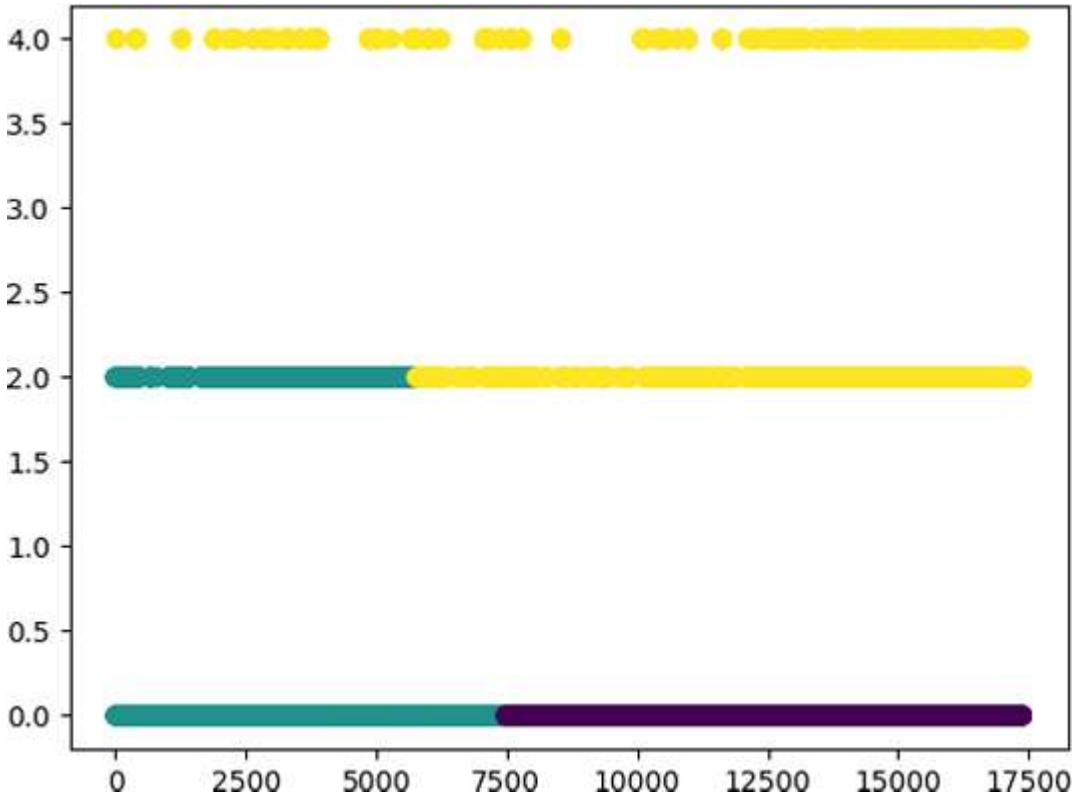
5. K-means Clustering:

- Groups data into clusters based on similarities, allowing easy identification of high-risk and low-risk periods or concentration zones for various particle sizes.

- Each cluster represents distinct behavior in particulate matter levels, providing actionable insights that are not immediately obvious in Random Forest or Polynomial Regression.

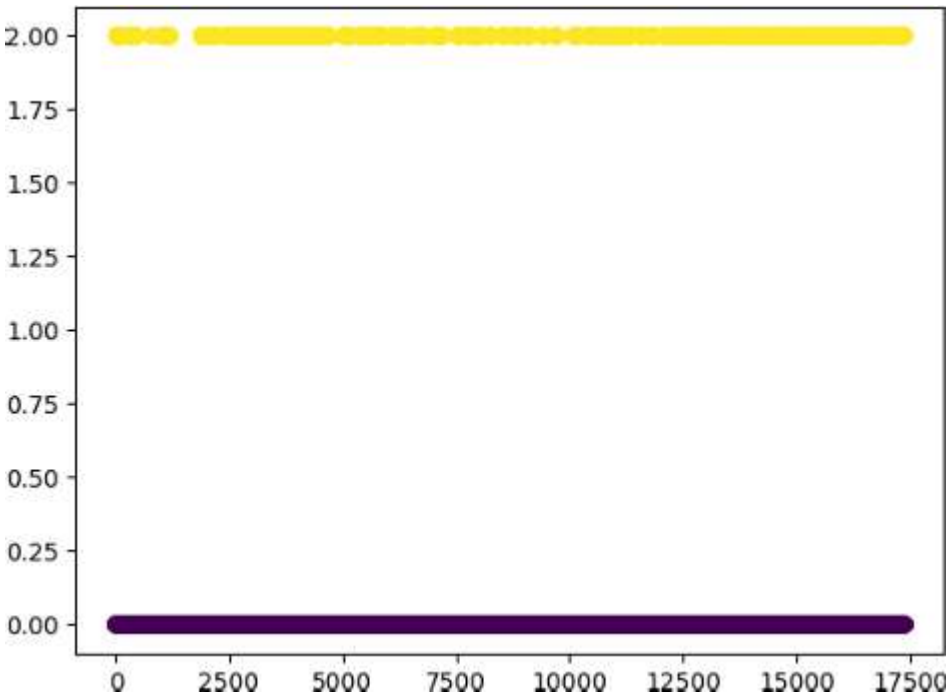
- Graphs (e.g., for 0.3µm, 0.5µm, PM1.0, PM2.5) show clear separations between clusters, highlighting trends in dust concentrations over time.





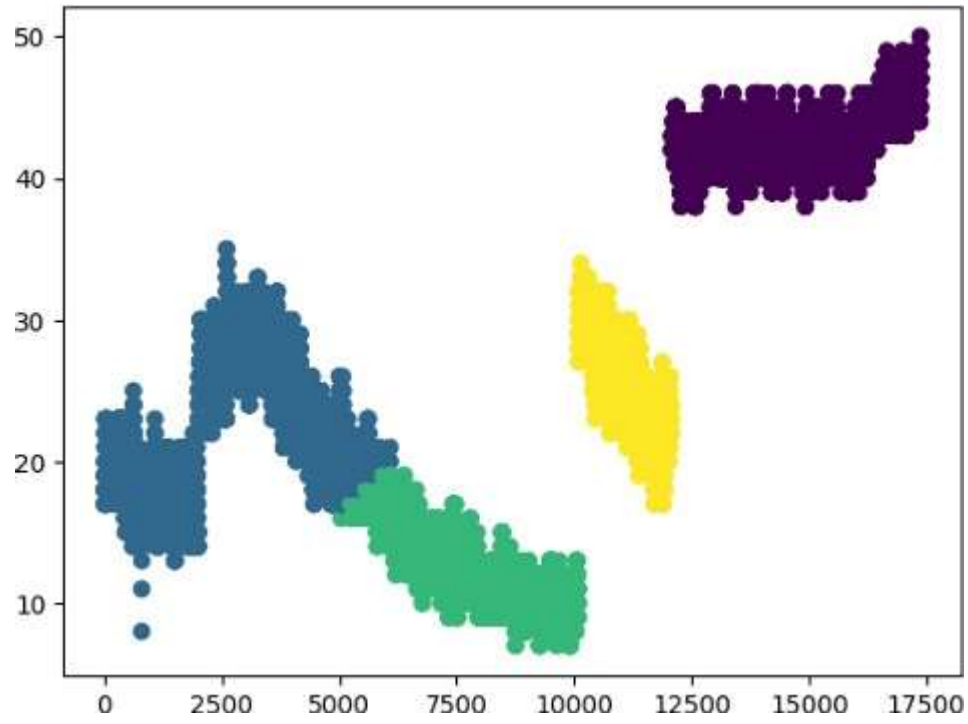
Graph for 5.0µm Particle Concentrations

- The clustering is less distinct, likely due to lower variability in 5.0µm data.
- Despite the data limitation, the clusters reveal general trends, although efficiency decreases for this particle size.



Graph for 10.0µm Particle Concentrations

- Clusters overlap more frequently, reflecting the limited granularity of 10.0µm data.
- The clustering is less effective due to insufficient data variability for larger particles.



Graph for PM1.0, PM2.5, and PM10

- Clusters are distinct for PM1.0 and PM2.5, highlighting clear trends and patterns.
- PM10 clustering shows reduced effectiveness due to larger particle variability.

6. Conclusion and Future Work: -

6.1 Summary: -

The project aims to predict particulate matter (PM) concentrations (PM1.0, PM2.5, PM10) using machine learning algorithms, leveraging IoT-enabled sensors for real-time monitoring. Dust pollution, particularly fine particles, poses significant health risks, necessitating accurate and proactive monitoring systems.

The system integrates the PMS7003 dust sensor, Arduino microcontroller, and TTL serial cable for data collection. Data is preprocessed to remove outliers, scale features, and structure it for analysis. Machine learning models, including Polynomial Regression and Random Forest Regression, were implemented to predict PM levels based on historical and real-time data. K-means clustering was used to identify patterns and high-risk zones by grouping similar concentration levels.

The results demonstrate high accuracy (~85%) for smaller particles like PM1.0 and PM2.5, with the model effectively capturing non-linear trends. However, predictions for larger particles (5.0µm and 10.0µm) were less accurate due to limited data variability. Visualization tools like scatter plots and regression curves showcased trends and clustering insights, enabling actionable responses.

This system enhances air quality monitoring, provides real-time alerts and preventive measures to reduce health risks. It aligns with sustainability goals, offering a scalable solution for urban planning and industrial applications, ultimately promoting healthier and safer environments.

6.2 Limitation and Constraints: -

The project successfully implements predictive analysis for particulate matter (PM) levels using machine learning and IoT, but it has certain limitations and constraints that impact its performance and scalability.

Limited Data Collection:
Data was collected for only 30 days and in a single location, restricting the model's ability to generalize across seasons, geographical regions, or varying environmental conditions.

Accuracy for Larger Particles:
The model performs poorly for larger particle sizes (5.0µm and 10.0µm), with accuracy as low as 5.43% and 2.19%, respectively. This is due to the limited variability and discrete nature of the data for these particles.

Sensitivity to Outliers:
Polynomial Regression is sensitive to outliers, which can distort the model's predictions. While preprocessing reduces some of this effect, it remains a challenge for long-term or diverse datasets.

Model Constraints:
Random Forest Regression requires significant computational resources for large datasets, and overfitting may occur if the number of trees is excessive.
K-means clustering assumes pre-defined clusters, which may not accurately represent real-world scenarios if dust patterns vary dynamically.

6.3 Improvement and Future Works: -

The current project demonstrates a robust framework for predictive analysis of particulate matter (PM) levels using IoT sensors and machine learning. However, several improvements and future directions can enhance its accuracy, scalability, and practical utility.

Expanded Data Collection:

- Collect data over a longer duration, covering different seasons and diverse geographic locations to improve the model's generalizability. Including high-dust environments such as construction sites or industrial zones will enhance the system's applicability.

Advanced Machine Learning Models:

- Experiment with neural networks or hybrid models to improve predictions for larger particles (e.g., 5.0μm and 10.0μm). These models can better capture complex patterns in diverse datasets.

Enhanced Real-time Capabilities:

- Integrate classification models to categorize dust levels into actionable categories like "safe," "moderate," and "hazardous." This would make the system more user-friendly for non-technical users.

Environmental Factors:

- Incorporate additional environmental variables, such as humidity, wind speed, and temperature, to improve the accuracy of PM predictions under varying conditions.

Mobile and Scalable Applications:

- Develop mobile or web-based platforms for real-time alerts and visualizations. This ensures wider accessibility for stakeholders, including industries, governments, and individuals.

Integration with Smart City Frameworks:

- Link the system with urban planning tools and smart city infrastructure to enable proactive air quality management, contributing to sustainable urban development.

7. SOCIAL AND ENVIRONMENTAL IMPACT

The project has significant social and environmental impacts, addressing critical health and sustainability challenges associated with dust pollution. By providing real-time monitoring and predictive analysis of particulate matter (PM) levels, it contributes to the well-being of individuals and the environment.

Social Impact:

Health Benefits:

- Real-time dust level monitoring and alerts help reduce exposure to harmful PM particles (PM1.0, PM2.5, PM10), which are linked to respiratory and cardiovascular diseases.
  Vulnerable populations, such as children, the elderly, and workers in high-dust environments, benefit from early warnings and preventive measures, improving overall public health.

Awareness and Education:

- The system raises awareness about air quality issues and encourages individuals to adopt protective measures, fostering a culture of environmental consciousness.
  Enhanced Decision-making:
  Policymakers and urban planners can use the insights generated by the system to implement air quality regulations and improve public spaces for cleaner living conditions.

Environmental Impact:

Pollution Reduction:

- By identifying high-risk areas and trends in dust concentration, the project supports targeted interventions to mitigate dust emissions from industrial activities, construction, and traffic.

Sustainability:

- The integration of IoT and machine learning technologies aligns with sustainable practices, promoting cleaner air and healthier ecosystems.

Climate Action:

- Reducing particulate matter in the atmosphere contributes to global climate goals, as high dust levels exacerbate climate change by influencing radiative forcing and cloud formation.

8. WORK PLAN

8.1 Timeline

The timeline for the project, covering the period from July 2024 to November 2024, is a follows:

July 2024

Week 1–2:

- Define the project scope and objectives.
  Identify the necessary hardware and software tools.
  Procure components (PMS7003 sensor, Arduino, TTL cable).

Week 3–4:

- Assemble the hardware setup and configure the PMS7003 dust sensor with Arduino.
  Test basic functionality to ensure accurate data collection.
  Write initial Arduino code to collect particulate matter data (PM1.0, PM2.5, PM10).

August 2024

Week 1–2:

- Begin data collection, saving raw sensor outputs in CSV format.
  Develop a data preprocessing pipeline to clean and prepare the dataset (e.g., handle NaN values, remove outliers).
  Explore sample visualizations for preliminary data insights.

Week 3–4:

- Train the Polynomial Regression model for PM level prediction.
  Validate initial predictions and tune the degree of the polynomial for optimal performance.

September 2024

Week 1–2:

- Train the Random Forest Regression model on the cleaned dataset.
- Evaluate the model using MSE, MAE, and R^2 scores for different PM sizes.

Week 3–4:

- Implement K-means clustering to group data points and identify dust concentration trends.
- Visualize clusters and analyze high-risk zones or periods.

October 2024

Week 1–2:

- Integrate all components for real-time predictions and alert generation.
- Test the complete system for continuous monitoring and prediction.

Week 3–4:

- Refine models and visualizations based on testing outcomes.
- Prepare draft documentation, including methodology and results analysis.

November 2024

Week 1–2:

- Finalize project report with conclusions, limitations, and future work.
- Prepare for project presentation, creating slides and live demonstrations.

Week 3–4:

- Conducted final testing and polish deliverables for submission.
- Submitted the completed project report and prepare for evaluations.

8.1 Individual Contribution

- Umar Khan

Hardware Setup:

Configured the PMS7003 sensor and Arduino microcontroller for data collection.
Ensured proper connectivity and testing of the TTL serial cable for seamless data transmission.

Data Collection:

Led the process of collecting particulate matter data (PM1.0, PM2.5, PM10) using the hardware setup.
Monitored the real-time sensor outputs and ensured accurate storage in CSV format.

Troubleshooting:

Addressed hardware-related challenges, such as sensor calibration and power supply issues.

- Aryan Mohapatra

Data Preprocessing:

Cleaned the collected data by handling NaN values and removing outliers using statistical methods.
Scaled and organized data for machine learning models, ensuring compatibility and accuracy.

Model Development:

Implemented Polynomial Regression for predicting dust levels.
Tuned the model's hyperparameters (degree of polynomial) to improve prediction accuracy.

Visualization:

Developed scatter plots, regression curves, and clustering visualizations using Matplotlib and Seaborn.

- Rohan Kumar

Machine Learning Implementation:

Trained and tested the Random Forest Regression model for particulate matter prediction.

Evaluated model performance using metrics like MSE, MAE, and R^2 scores.

Clustering Analysis:

Applied K-means clustering to group similar dust concentrations, identifying high-risk patterns and zones.

Interpreted clustering results to derive actionable insights for dust level management.

Documentation:

Contributed to drafting the project report, including the methodology, results, and limitations.

9. COST ANALYSIS:

- Arduino Uno:

Cost: ₹800 – ₹1,000

Purpose: Microcontroller used to interface with the PMS7003 sensor for data collection and transmission.

- TTL Serial Download Cable:

Cost: ₹400 – ₹600

Purpose: Connects the Arduino to the computer for data transfer and sensor programming.

- Miscellaneous Hardware (Wires, Connectors, Breadboard, Power Adapter):

Cost: ₹300 – ₹500

Purpose: Ensures proper connectivity and operation of the hardware setup.

Software Tools

Arduino IDE:
Cost: Free (Open-source)
Purpose: Programming and configuring the Arduino microcontroller.

Python (with Libraries: Pandas, NumPy, sklearn, Matplotlib):
Cost: Free (Open-source)
Purpose: Data preprocessing, analysis, machine learning model implementation, and visualization.

ArduSpreadsheet:
Cost: Free
Purpose: Converts Arduino serial output to CSV format for data storage and processing.
Miscellaneous Costs

Printing and Documentation:
Cost: ₹200 – ₹300
Purpose: Covers printing the final report and presentation materials.

10. References

- [1] Xiong, Ruoxin & Tang, Pingbo. (2021). Machine learning using synthetic images for detecting dust emissions on construction sites. Smart and Sustainable Built Environment. ahead-of-print. 10.1108/SASBE-04-2021-0066.
- [2] Machine learning holography for measuring 3D particle distribution by Siyao Shao
- [3] Machine learning shadowgraph for particle size and shape characterization by Jiaqi Li
- [4] A Survey paper on Vehicles Emitting Air Quality and Prevention of Air Pollution by using IoT Along with Machine Learning Approaches by M.Dhanalakshmi.
- [5] Méndez, M., Merayo, M.G. & Núñez, M. Machine learning algorithms to forecast air quality: a survey. Artif Intell Rev (2023). https://doi.org/10.1007/s10462-023-10424-4
- Pedersini, F. (2018). Improving a Commodity Dust Sensor to Enable Particle Size Analysis. IEEE Transactions on Instrumentation and Measurement, 1–12. doi:10.1109/tim.2018.2834178
- Proietti, A., Panella, M., Leccese, F., & Svezia, E. (2015). Dust detection and analysis in museum environment based on pattern recognition. Measurement, 66, 62–72. doi:10.1016/j.measurement.2015.01
- Liu, D., Zhao, W., Li, D., Wang, J., & Dong, C. (2019). Optimization of dust concentration measuring device. IOP Conference Series: Materials Science and Engineering, 592, 012184. doi:10.1088/1757
- Chen, L., Zhu, D., Tian, J., & Liu, J. (2016). Dust particle detection in traffic surveillance video using motion singularity analysis. Digital Signal Processing, 58, 127–133. doi:10.1016/j.dsp.2016.07.020
- Srama, R., & Grün, E. (1997). The dust sensor for CASSINI. Advances in Space Research, 20(8), 1467–1470. doi:10.1016/s0273-1177(97)00418-3