AirCo
Real Time Air Quality Prediction and Data Analysis

MINI PROJECT REPORT
Submitted by,

AFNA V.N (ASI22CA007)
JEFFY JOHSON(ASI22CA027)
JOFIN JOJI (ASI22CA028)
K VAISHNAV NAIR(ASI22CA034)

Under the guidance of

ASST.PROF MEENATCHI K.V

To

APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree of
Bachelor of Technology in Computer Science and Engineering (AI)

Department of Artificial Intelligence and Data Science

CSE(AI) 2022-26 BATCH

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY
Kalady- 683574

April 2025

AirCo
Real Time Air Quality Prediction and Data Analysis

MINI PROJECT REPORT
Submitted by,

AFNA V.N (ASI22CA007)
JEFFY JOHSON(ASI22CA027)
JOFIN JOJI (ASI22CA028)
K VAISHNAV NAIR(ASI22CA034)

Under the guidance of

ASST.PROF MEENATCHI K.V

To

APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree of
Bachelor of Technology in Computer Science and Engineering (AI)

Department of Artificial Intelligence and Data Science

CSE(AI) 2022-26 BATCH

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY
Kalady- 683574

April 2025

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY
Kalady- 683574

Department of Artificial Intelligence and Data Science

CERTIFICATE

Certified that this is a bonafide record of the project entitled

AirCo
Real Time Air Quality Prediction and Data Analysis

Submitted by

Afna V.N
ASI22CA007
Jeffy Johnson
ASI22CA027
Jofin Joji
ASI22CA028
K Vaishnav Nair
ASI22CA034

during the year 2024-25 in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering (AI)

Project Guide	Project Coordinator	Head of the Department

ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY

Kalady- 683574

Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the project entitled **AirCo Real Time Air Quality Prediction and Data Analysis** has been successfully carried out by **Afna V.N ASI22CA007, Jeffy Johnson ASI22CA027, Jofin Joji ASI22CA028, K Vaishnav Nair ASI22CA034** in partial fulfillment of the course Bachelor of Technology.

Date: **HEAD OF DEPARTMENT**

DECLARATION

We ,Afna V.N, Jeffy Johnson, Jofin Joji, K Vaishnav Nair, hereby declare that the project work entitled **AirCo Real Time Air Quality Prediction and Data Analysis**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of **Asst.Prof Meentachi K.V**. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained.

PLACE: KALADY

DATE:

SIGNATURE

Afna V.N

Jeffy Johnson

Jofin Joji

K Vaishnav Nair

ACKNOWLEDGMENT

In the name of almighty **GOD**, I express my sincere thanks to him keeping me fit for successful completion of the project.

I am thankful to **Dr. M S MURALI, Principal, ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY** for his kind support in all respect during our study.

I sincerely extend our thanks to our **CTO, Prof. P V RAJARAMAN**, for all the help, motivation and guidance to carry out this project.

I sincerely thank **Dr. SARIKA S, HOD, Department of Artificial Intelligence and Data Science, ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY**, for her encouragement to carry out this project.

I especially thank my Guide **ASST.PROF MEENTACHI K.V**, Department of Artificial Intelligence and Data Science, **ADI SHANKARA INSTITUTE OF ENGINEERING AND TECHNOLOGY**, in examining the draft of this seminar and for valuable suggestions.

I want to thank the Department of Artificial Intelligence and Data Science for giving me the permission to prepare the project on the topic **AirCo Real Time Air Quality Prediction and Data Analysis.**

Afna V.N

Jeffy Johnson

Jofin Joji

K Vaishnav Nair

ABSTRACT

This project presents an IoT-based air quality monitoring and prediction system that utilizes machine learning techniques to deliver real-time insights and forecast air pollution trends. The system integrates a network of IoT sensors to measure pollutants such as PM2.5, PM10, NO2, CO, and NH3, alongside environmental parameters like temperature, humidity, and pressure. The collected data is transmitted to a cloud-based platform for secure storage and processing. Advanced machine learning algorithms, including linear regression, decision trees, and Long Short-Term Memory (LSTM) networks, analyse both historical and real-time data to predict future air quality levels with improved accuracy.

A web-based dashboard serves as the primary user interface, offering intuitive visualizations of live pollutant concentrations, historical trends, and forecasted Air Quality Index (AQI) values. Users, ranging from policymakers and environmental researchers to the general public, can access these insights to take preventive measures against exposure to harmful pollutants. Additionally, the system features an automated alert mechanism that notifies users via email when air quality deteriorates beyond safe thresholds.

The architecture emphasizes scalability and accessibility, enabling the deployment of multiple sensor nodes across different locations, while cloud integration ensures efficient data management and retrieval. The system also supports role-based access to data, catering to diverse user groups with specific analytical needs. By leveraging IoT and machine learning, this solution aims to bridge the gap between traditional air quality monitoring methods and modern data-driven approaches, empowering communities to make informed decisions and promoting sustainable urban development.

CONTENTS

LIST OF FIGURES

LIST OF TABLE

ABBREVIATION

General Abbreviations

- AI – Artificial Intelligence
- ML – Machine Learning
- IoT – Internet of Things
- SDLC – Software Development Life Cycle
- API – Application Programming Interface
- DL – Deep Learning

Air Quality & Environmental Parameters

- PM2.5 – Particulate Matter (diameter $\leq$ 2.5 microns)
- PM10 – Particulate Matter (diameter $\leq$ 10 microns)
- NO2 – Nitrogen Dioxide
- CO – Carbon Monoxide
- NH3 – Ammonia
- AQI – Air Quality Index

Technical Terms & Frameworks

- LSTM – Long Short-Term Memory (Neural Network)
- RF – Random Forest (Machine Learning Algorithm)
- MSE – Mean Squared Error (Loss Function)
- R^2 – R-squared (Regression Performance Metric)
- Flask – Python Web Framework
- SQL – Structured Query Language
- API – Application Programming Interface
- SMTP – Simple Mail Transfer Protocol
- SSL – Secure Sockets Layer
- HTTP – Hypertext Transfer Protocol
- ESP32 – IoT Microcontroller

Cloud & Storage

- AWS – Amazon Web Services
- GCP – Google Cloud Platform
- DB – Database
- CSV – Comma-Separated Values

Chapter 1
Introduction

1. INTRODUCTION

1.1 Background

Air pollution is a growing global concern, with rising levels of pollutants like PM2.5, PM10, NO2, CO, and NH3 posing serious health and environmental risks. Traditional monitoring systems, often limited in coverage and costly, fail to provide real-time insights across diverse locations.

The integration of IoT and machine learning offers a promising solution. IoT sensors enable continuous, real-time pollutant tracking, while machine learning algorithms analyze data to predict future air quality trends. This project aims to develop a system that combines these technologies, providing timely alerts and empowering policymakers, researchers, and the public with data-driven insights to mitigate pollution impacts and promote healthier environments.

1.2 Existing System

Traditional air quality monitoring systems rely on government-operated stations equipped with high-precision sensors to track pollutant levels. While these stations provide reliable data, their high cost and limited number result in sparse coverage, especially in densely populated or remote areas. Additionally, data updates are often delayed, making real-time monitoring challenging.

In recent years, IoT-based systems have emerged to address these limitations, offering affordable sensor networks that enable continuous data collection across multiple locations. However, most existing IoT solutions focus primarily on data collection and visualization, lacking advanced predictive models to forecast air quality trends accurately. Furthermore, accessibility remains limited, with few systems providing user-friendly dashboards or timely alerts.

This project aims to bridge these gaps by combining IoT-based real-time monitoring with machine learning-driven predictions, delivering comprehensive analysis, accurate forecasting, and email alerts through a web-based platform.

1.3 Problem Statement

Air pollution poses significant health and environmental risks, yet existing air quality monitoring systems face several limitations. Traditional monitoring stations are costly to install and maintain, leading to sparse coverage and delays in data availability. As a result, communities lack real-time insights into pollution levels and early warnings to take preventive actions.

While IoT-based monitoring solutions offer wider coverage and continuous data collection, many existing systems focus only on pollutant tracking, lacking accurate prediction mechanisms and user-friendly access to data. Additionally, timely alerts for deteriorating air quality are often absent, reducing the effectiveness of such systems in safeguarding public health.

This project aims to develop an IoT-based air quality monitoring and prediction system that addresses these challenges. By integrating real-time pollutant tracking with machine learning models, the system provides accurate forecasts and timely email alerts, empowering users to make informed decisions and take proactive measures against pollution exposure.

1.4 Scope

Air pollution is a major environmental and health concern, with pollutants like PM2.5, PM10, NO2, CO, and NH3 affecting millions of people worldwide. Traditional monitoring stations provide accurate data but are costly and limited in coverage. To overcome these challenges, this project develops an IoT-based air quality monitoring and prediction system using machine learning techniques for real-time insights and forecasting.

The system integrates IoT sensors that continuously measure air quality parameters and transmit data to a cloud-based storage system. This ensures real-time monitoring and remote accessibility for users. The collected data is processed using machine learning models, including LSTM neural networks, decision trees, and random forests, to predict future air quality trends.

A web-based dashboard serves as the primary interface, displaying live air quality readings, historical trends, and AQI forecasts. To enhance user awareness, the system includes an automated alert mechanism that notifies users via email when pollution levels exceed safe thresholds.

By combining IoT, cloud computing, and machine learning, this system provides an affordable, scalable, and data-driven solution for air quality monitoring, helping communities and policymakers take proactive measures to reduce pollution exposure.

Chapter 2
LITERATURE SURVEY

2. LITERATURE SURVEY

No.	Title	Authors	Year	Methodology	Metrics & Accuracy	Dataset	Advantages	Disadvantages
1	Air Quality Monitoring System	Ramik Rawal	Jan 2019	Uses MQ135 & MQ7 sensors with Arduino Uno & ESP-01 WiFi module; Data transmitted to Thingspeak; Threshold-based alert system	PPM, Sensor output voltage, Response time, Network latency; Accuracy: ±5-10%	CO, CO2, NH3, NOx, Benzene, Smoke, Alcohol; Time-series & location-based data stored in Things peak	Real-time monitoring, Cost-effective, Alerts, Scalable	Calibration needed, WiFi dependency, Limited sensor range, Power consumption issues, Environmental factors affect accuracy
2	Artificial Intelligence for Air Quality and Control Systems	Divya Patel, Mridu Kulwant, Saba Shirin, Ankit Kumar, Mohammad Aurangzeb Ansari, Akhilesh Kumar Yadav	Jul 2022	AI-based prediction using Neural Networks (ANNs), LSTM, GRU, SVR, ARIMA, Random Forest, Logistic Regression	AQI, RMSE, Index of Agreement; High accuracy through ensemble techniques	Pollutant levels (PM2.5, PM10, SO_2, NO_2, CO, O_3), Meteorological data	Real-time monitoring, Early warnings, High accuracy, Automation, AI integration	Data dependency, Computational complexity, Black-box nature, High implementation cost
3	Research on	Zijie Liu, Kaijie	Sep 2023	Uses Pearson correlation, Stepwise	Predicted R^2: 0.905,	PM2.5, AQI, Meteor	High accuracy, Strong	Feature selection limits, Limited

	Air Quality Prediction based on Correlation Analysis and XGBoost	Chen, Ziyu Ning, Lele Wang, Zhaoting Zheng		regression, Chi-square test, XGBoost model; Compares with SVM & BP Neural Networks	Feature importance; High accuracy	ological factors	feature extraction, Advanced ML, Better than traditional models, Public health relevance	dataset details, Temporal/spatial factors, Real-world implementation concerns
4	Revolutionizing Air Quality Forecasting with ML & Optimization	Krati Dubey, Rishav Dubey, Saroj Pandey, Anurag Sinha, Yogesh Rahhore	Jul 2024	Uses LSTM, CNNs, GBMs, Statistical models, Optimization techniques; Comparative analysis with traditional methods	Prediction accuracy, Computational efficiency, RMSE, MAE, R^2; Accuracy not mentioned	Not provided	High accuracy, Optimized models, Scalable, Computationally efficient	Data dependency, Complexity of atmosphere, High resource requirements, Model interpretability, Scalability concerns
5	Monitoring & Predicting Air Quality with IoT Devices	Claudia Banciu, Adrian Florea, Răzvan Bogdan	Sep 2024	IoT-based data collection (5869 data points); Uses Feedforward NN, Random Forest Regression analysis	MAE (AQI 10: 0.2785, AQI 2.5: 0.2483), Model comparison; High accuracy	5869 data points (Temp, Humidity, PM10, PM2.5) stored in Things peak	Real-time monitoring, Identifies pollution hotspots, Advanced ML, IoT integration, Public health applications	Data dependency, Sensor accuracy, Model complexity, Limited parameters, Latency in real-time processing

Table 1: Literature Survey

Air pollution remains a critical environmental and health challenge, requiring advanced monitoring and predictive techniques to mitigate its impact. Various studies have explored the use of IoT, machine learning, and artificial intelligence to enhance air quality monitoring and forecasting accuracy.

Early studies, such as Rawal (2019), proposed IoT-based air quality monitoring systems using sensors like MQ135 and MQ7. These systems effectively measure CO, CO_2, NH_3, NO_x, and other pollutants, transmitting data to cloud platforms like ThingSpeak for real-time visualization. However, challenges such as sensor calibration, WiFi dependency, and environmental variations affecting accuracy were identified.

Advancements in machine learning (ML) and artificial intelligence (AI) have significantly improved air quality prediction models. Patel et al. (2022) applied Artificial Neural Networks (ANNs), LSTM, GRU, SVR, ARIMA, and Random Forest Regression for AQI forecasting. Their integration of meteorological data enhanced prediction accuracy, enabling real-time monitoring and early warnings. However, AI-based systems often suffer from high computational costs and black-box model interpretability issues.

Liu et al. (2023) introduced correlation analysis and XGBoost-based models to improve feature selection and accuracy in air quality predictions. Their study demonstrated higher accuracy ($R^2 = 0.905$) compared to SVM and BP Neural Networks, proving the effectiveness of advanced ML techniques. However, limitations in feature selection and dataset availability pose challenges for real-world implementation.

Dubey et al. (2024) explored deep learning approaches, combining LSTM, CNN, and GBMs with optimization techniques for enhanced air quality forecasting. Their approach improved computational efficiency and model scalability, yet the complexity of atmospheric conditions and data dependency remain key concerns.

Banciu et al. (2024) emphasized IoT-driven real-time air quality monitoring, integrating Feedforward Neural Networks and Random Forest Regression. Their system effectively identifies pollution hotspots and supports public health interventions. However, sensor accuracy, real-time latency, and model complexity still pose challenges.

Overall, the combination of IoT, cloud computing, and AI-driven predictive analytics offers a promising solution for real-time air quality monitoring and forecasting. Future improvements should focus on enhancing model interpretability, increasing data availability, and optimizing computational efficiency to ensure widespread adoption and practical implementation.

2.1 Air Quality Monitoring System (2019) – Ramik Rawal

Methodology

This research presents an IoT-based air quality monitoring system designed to track pollution levels in real time. The system uses MQ135 and MQ7 gas sensors to measure air pollutants such as carbon monoxide (CO), carbon dioxide (CO_2), ammonia (NH_3), nitrogen oxides (NO_x), benzene, and smoke. These sensors are integrated with an Arduino Uno microcontroller, which processes the sensor readings. The collected data is transmitted to an IoT platform, specifically Thingspeak, using an ESP-01 WiFi module.

The system includes a calibration mechanism to convert raw sensor outputs into PPM (Parts Per Million) values for accurate air quality measurement. A threshold-based alert system is implemented to notify users when pollutant concentrations exceed safe limits. The primary goal of the study is to provide an affordable and scalable solution for continuous air quality monitoring.

Dataset

The dataset consists of time-series air quality data collected from the MQ135 and MQ7 sensors. The values are recorded in real-time and stored on Thingspeak, making the data accessible remotely. The
system can be deployed in multiple locations to compare pollution levels in different environments.

Metrics and Accuracy

The performance of the system is evaluated using the following metrics:

- PPM (Parts Per Million) values to measure pollutant concentration.
- Sensor output voltage conversion to ensure data accuracy.
- Response time indicating how quickly the system detects pollution changes.
- Network latency, affecting real-time data transmission.
- The system maintains an accuracy of approximately ±5-10%, depending on calibration and environmental conditions.

Advantages

The system is highly cost-effective, making it an accessible solution for widespread air quality monitoring. The use of IoT enables real-time data transmission and visualization on a public dashboard. The system is scalable and can be implemented in different locations for comparative analysis. The alert mechanism allows immediate responses to hazardous pollution levels.

Disadvantages

The system requires regular calibration to maintain measurement accuracy. Since it relies on WiFi for data transmission, it may experience connectivity issues in areas with unstable networks. The sensors used have a limited detection range, making them less effective for large-scale monitoring. Environmental factors such as temperature and humidity fluctuations can affect sensor accuracy.

2.2 Artificial Intelligence for Air Quality and Control Systems (2022) – Divya Patel et al.

Methodology

This study applies Artificial Intelligence (AI) and Machine Learning (ML) techniques to analyze and predict air quality. Several models are used, including Artificial Neural Networks (ANNs), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Support Vector Regression (SVR), Autoregressive Integrated Moving Average (ARIMA), and Random Forest Regression.

The methodology focuses on integrating both historical and real-time air quality data with meteorological parameters. The AI models analyze pollutant trends and forecast future pollution levels. The inclusion of meteorological data (wind speed, temperature, and humidity) enhances prediction accuracy by accounting for environmental factors affecting pollutant dispersion.

Dataset

The dataset is collected from Continuous Ambient Air Quality Monitoring Stations (CAAQMS). It includes pollutant concentrations of PM2.5, PM10, SO_2, NO_2, CO, and O_3, along with meteorological variables such as temperature and humidity.

Metrics and Accuracy

The study evaluates model performance using:

- Air Quality Index (AQI) to measure pollution severity.
- Root Mean Square Error (RMSE) for prediction accuracy.
- Index of Agreement (IA) to assess model performance.
- High accuracy achieved through ensemble learning techniques.

Advantages

The system enables real-time air quality forecasting, allowing early warnings for pollution hazards. The use of advanced AI models improves accuracy over traditional statistical approaches. Automation reduces manual monitoring efforts, making the system efficient for large-scale deployment.

Disadvantages

The system is highly dependent on high-quality datasets, making it sensitive to missing or biased data. The use of complex AI models requires significant computational power, limiting deployment in resource-constrained environments. The AI models operate as black-box systems, making it difficult to interpret how predictions are made. Additionally, setting up AI-based air monitoring systems requires substantial financial investment.

2.3 Research on Air Quality Prediction using XGBoost (2023) – Zijie Liu et al.

Methodology

The study utilizes correlation analysis and the XGBoost machine learning model for air quality prediction. The Pearson correlation model and stepwise regression model are used for quantitative feature extraction, while the chi-square test is applied for categorical features. The XGBoost model is chosen for its efficiency in handling large datasets and providing high-accuracy predictions.

The methodology includes a comparative analysis with Support Vector Machines (SVM) and BP Neural Networks, demonstrating XGBoost's superior performance in handling non-linear relationships in air quality data.

Dataset

The dataset focuses on PM2.5 concentration and the Air Quality Index (AQI). Meteorological parameters such as temperature, humidity, and wind speed are also included to improve prediction accuracy. The study does not specify the exact dataset used, raising concerns about data generalizability.

Metrics and Accuracy

- The model achieves a predicted R^2 value of 0.905, indicating high correlation with actual values.
- The study confirms that XGBoost outperforms traditional models such as SVM and BP neural networks in terms of prediction accuracy.

Advantages

The system demonstrates high prediction accuracy and effectiveness in feature selection. It outperforms traditional statistical models and is highly relevant for public health monitoring and policymaking.

Disadvantages

The study does not provide details on data sources, making it difficult to validate results. Some important air quality factors may not have been included in feature selection, potentially affecting prediction accuracy.

2.4 Revolutionizing Air Quality Forecasting with ML & Optimization (2024) – Krati Dubey et al.

Methodology

This research integrates deep learning and statistical optimization techniques for air quality forecasting. The models used include LSTM for time-series prediction, CNN for spatial pattern recognition, and Gradient Boosting Machines (GBMs) for ensemble learning.
Optimization strategies are employed to refine model performance, improving both accuracy and computational efficiency. The study also compares AI-based predictions with traditional forecasting methods, demonstrating the superiority of machine learning approaches.

Dataset

The study does not provide details about the dataset used.

Metrics and Accuracy

- The study uses Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 as performance metrics.
- The exact accuracy percentage is not provided, but results indicate improved performance over conventional methods.

Advantages

The ensemble approach significantly enhances prediction accuracy and computational efficiency. The framework is designed to be scalable, making it applicable to various environmental monitoring scenarios.

Disadvantages

The study is highly dependent on data quality and requires substantial computational resources. The complexity of deep learning models reduces interpretability, making it difficult for policymakers to trust predictions.

2.5 Monitoring and Predicting Air Quality with IoT (2024) – Claudia Banciu et al.

Methodology

The study employs IoT-based sensors to collect real-time data on PM10, PM2.5, temperature, and humidity. Data is stored in ThingSpeak, and machine learning models including Random Forest Regression and Feedforward Neural Networks are used for AQI prediction.

Dataset
The dataset consists of 5869 data points collected from IoT devices, covering various environmental parameters.

Metrics and Accuracy
- Mean Absolute Error (MAE): AQI 10 = 0.2785, AQI 2.5 = 0.2483.
- The Random Forest model with 100 estimators achieves the highest accuracy.

Advantages
The study enables real-time air quality monitoring, identifies pollution hotspots, and supports public health interventions.

Disadvantages
Sensor calibration is required, and the system faces latency issues in real-time data processing.

Chapter 3
PROPOSED SYSTEM

3. PROPOSED SYSTEM

3.1 Objective

The objective of this project is to develop an IoT-based air quality monitoring and prediction system that offers real-time insights and accurate forecasts of pollution levels. The system aims to continuously measure pollutants such as PM2.5, PM10, NO_2, CO, and NH_3, along with environmental parameters like temperature, humidity, and pressure, using IoT sensors. Machine learning algorithms are applied to analyse the collected data and predict future air quality trends. To ensure timely awareness, automated email alerts are sent when pollutant levels exceed safe thresholds. Additionally, a user-friendly web-based dashboard provides real-time visualizations, historical data analysis, and predictive insights, enabling users to make informed decisions. By integrating these components, the project strives to create a cost-effective, scalable, and accessible solution to enhance air quality monitoring and public health awareness.

3.2 Methodology

The development of this IoT-based air quality monitoring and prediction system involves several key stages, integrating real-time data collection, machine learning-based analysis, and user-friendly visualization.

1. **Data Collection**
   IoT sensors are deployed to measure pollutant concentrations, including PM2.5, PM10, NO_2, CO, NH_3, and environmental parameters like temperature, humidity, and pressure. These sensors are connected to microcontrollers such as Arduino or ESP32, which transmit the collected data to a cloud-based storage system at regular intervals.

2. **Data Storage and Preprocessing**
   The raw sensor data is stored in a cloud database (e.g., Firebase or MySQL) for further processing. Data preprocessing involves handling missing values, normalizing pollutant concentrations, and integrating meteorological data to improve prediction accuracy.

3. **Machine Learning Model Training**
   Historical air quality data is used to train machine learning models such as linear regression, decision trees, and Long Short-Term Memory (LSTM) networks. These models analyse trends and patterns in the data, predicting future air quality levels based on real-time inputs. Model performance is evaluated using metrics like Mean Squared Error (MSE) and R^2 to ensure accuracy.

4. **Real-Time Prediction and Visualization**
   As new data is collected, the trained models make real-time predictions of pollutant levels and Air Quality Index (AQI). A web-based dashboard is developed to display live air quality readings, historical trends, and forecasted values, ensuring accessibility for users.

5. **Alert Mechanism**
   The system continuously monitors air quality and sends automated email alerts to users

when pollutant levels exceed predefined safety thresholds, allowing timely preventive action.

6. **System Maintenance and Updates**
Regular calibration of sensors is performed to ensure data accuracy. Machine learning models are periodically retrained with new data to maintain prediction reliability. The system's performance and scalability are monitored to support deployment across multiple locations.

3.2.1 System Architecture

Figure 1 illustrates the System architecture

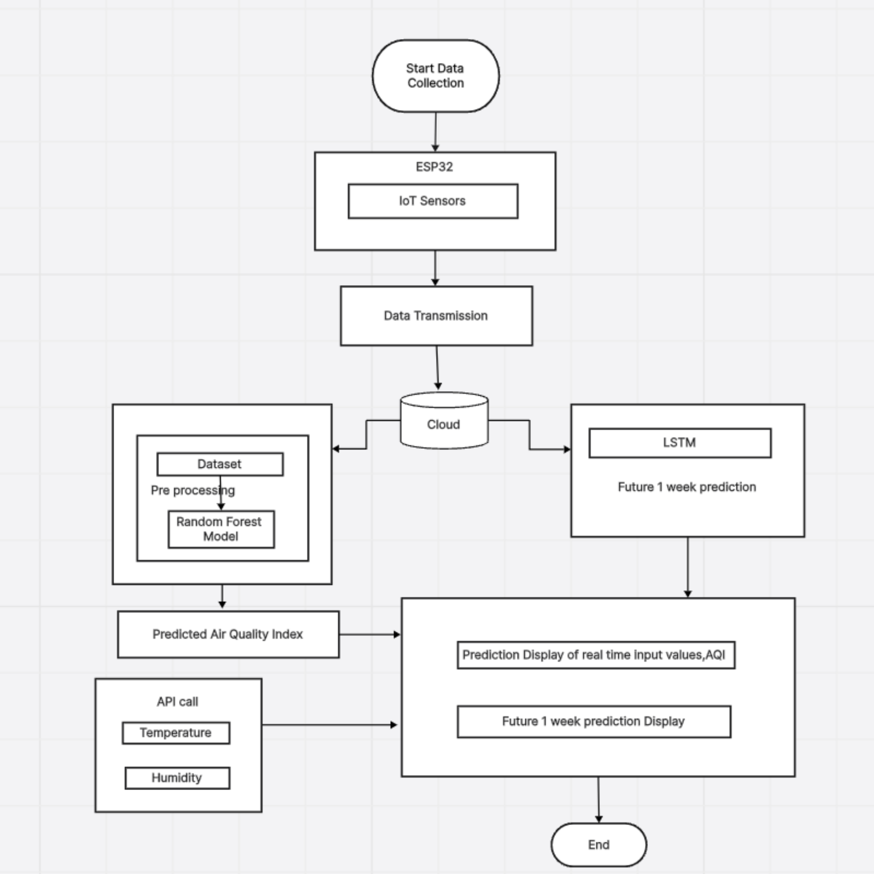

Fig 1 Architecture

3.2.2 Flow Diagram

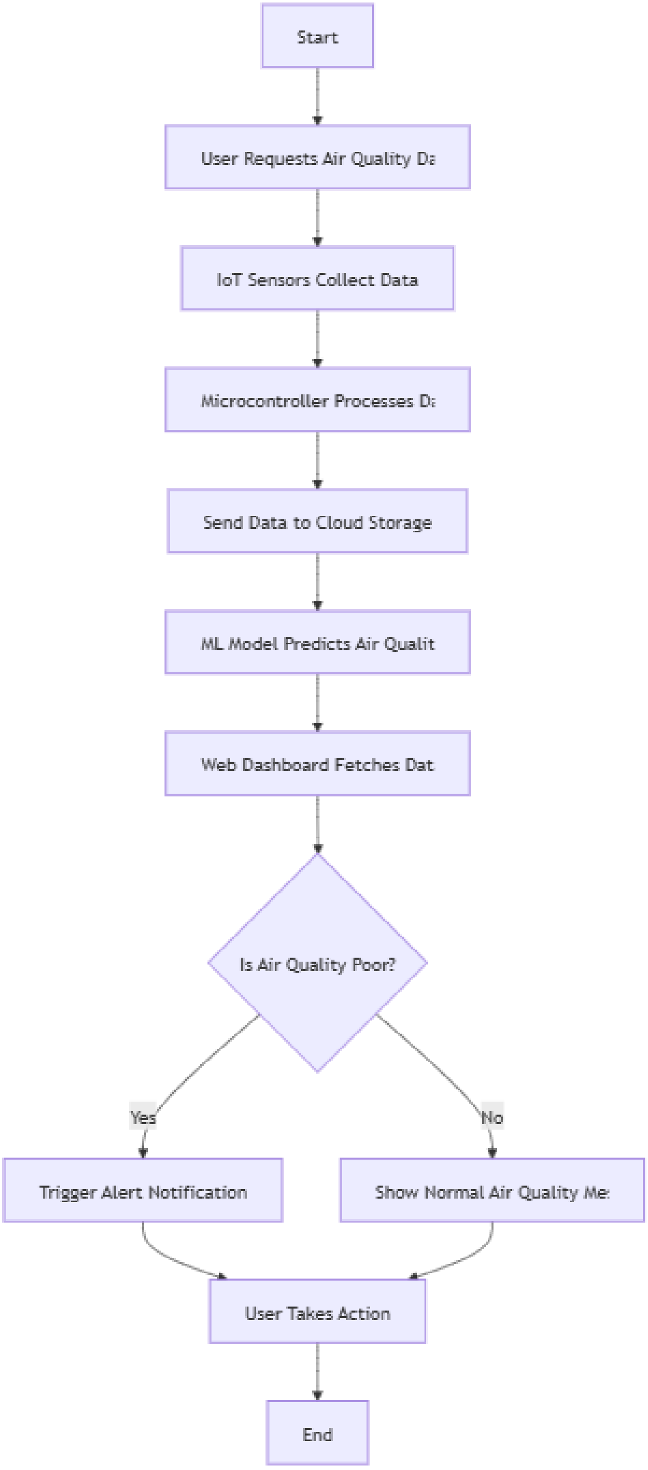

Fig 2 Flow Diagram

3.2.3 Functional Requirements

- Collect data from IoT sensors (PM2.5, PM10, NO2, CO, NH3, Temperature, Humidity, Pressure). Store and manage collected data in a cloud database.
- Train ML models (Linear Regression, Decision Trees, LSTM) to predict air quality.
- Display real-time and forecasted air quality levels on a website.
- Provide alerts for poor air quality.
- Ensure scalability for multiple sensor networks and locations.

3.2.4 Non – Functional Requirements

The system must ensure reliability through continuous data collection and processing, minimizing downtime and maintaining accurate pollutant measurements. It should be scalable, supporting the integration of additional sensors and new locations without impacting performance. Real-time processing is essential, with minimal latency to deliver timely insights and alerts. Security measures must be in place, ensuring encrypted data transmission and access control to protect sensitive information. The web-based dashboard should prioritize usability, offering an intuitive interface for accessing real-time data, historical trends, and predictive insights. Maintainability is crucial, with a modular design that allows easy updates to sensors, machine learning models, and the dashboard. The system should also be portable, ensuring compatibility with various IoT devices and adaptability across different environments. Environmental resilience is necessary, with sensors capable of withstanding diverse conditions such as temperature fluctuations and humidity. Data integrity mechanisms should handle missing data, detect anomalies, and ensure the accuracy of pollutant levels. Finally, the system must comply with regulatory standards for air quality monitoring and data privacy, ensuring its alignment with established guidelines.

3.2.5 Hardware Requirements

- Microcontroller: Arduino, Raspberry Pi, or ESP32.
- Air Quality Sensors: MQ Series, PMS5003.
- Power Supply: Batteries or adapters.
- Cloud or local server for storing and processing data.

3.2.6 Software Requirements

- Programming: Python, C++/Arduino, Flask.
- ML Libraries: Scikit-learn, Pandas, NumPy, Matplotlib.
- DL Libraries: Tensor Flow, Keras
- IoT Platforms: ThingSpeak, Node-RED.
- Database: Google Sheets.
- Web-based Dashboard: JavaScript, HTML, CSS, Flask.

3.2.7 Life Cycle Used

For the IoT-based Air Quality Monitoring and Prediction System, the Agile Software Development Life Cycle (SDLC) model is the most suitable approach. This model is ideal because IoT and machine learning-based systems require continuous updates, iterative improvements, and real-time feedback. The project undergoes several phases, starting with Requirement Analysis, where system goals, sensor selection, cloud integration, and machine learning algorithms are defined. Next, in the Planning and Design phase, the architecture of the system is structured, including microcontroller-sensor interactions, data storage mechanisms, and dashboard UI design. Once the planning is complete, the Implementation phase involves coding for IoT sensor data collection, cloud database integration, and the development of machine learning models for air quality prediction. After the initial development, the Testing and Deployment phase ensures that real-world sensor data is accurately processed, the ML models provide reliable predictions, and the system delivers real-time alerts for poor air quality. Finally, the Monitoring and Maintenance phase focuses on updating machine learning models based on new environmental data trends, recalibrating sensors, and refining the user experience on the dashboard. This Agile approach allows for continuous refinement based on user feedback, ensuring that the system adapts to changing environmental factors and technological advancements. Additionally, it provides flexibility in integrating new features such as improved ML models, expanded sensor networks, or additional alert mechanisms. By using Agile, the project can evolve dynamically, ensuring high accuracy in air quality monitoring and predictions.

3.2.8 Economic Feasibility

This IoT-based air quality monitoring system is designed to be cost-effective by leveraging affordable hardware and open-source software. Components like ESP32 microcontrollers and MQ-series sensors are inexpensive and easy to source, while cloud-based storage eliminates the need for costly on-site servers. The use of open-source libraries for machine learning further reduces software expenses. Additionally, the system's scalability allows new sensors to be added without altering the core infrastructure, ensuring long-term affordability. Overall, the project offers a financially viable solution for real-time air quality monitoring with minimal maintenance costs and high scalability.

Chapter 4
System Design

4. SYSTEM DESIGN

4.1 Use Case Diagram

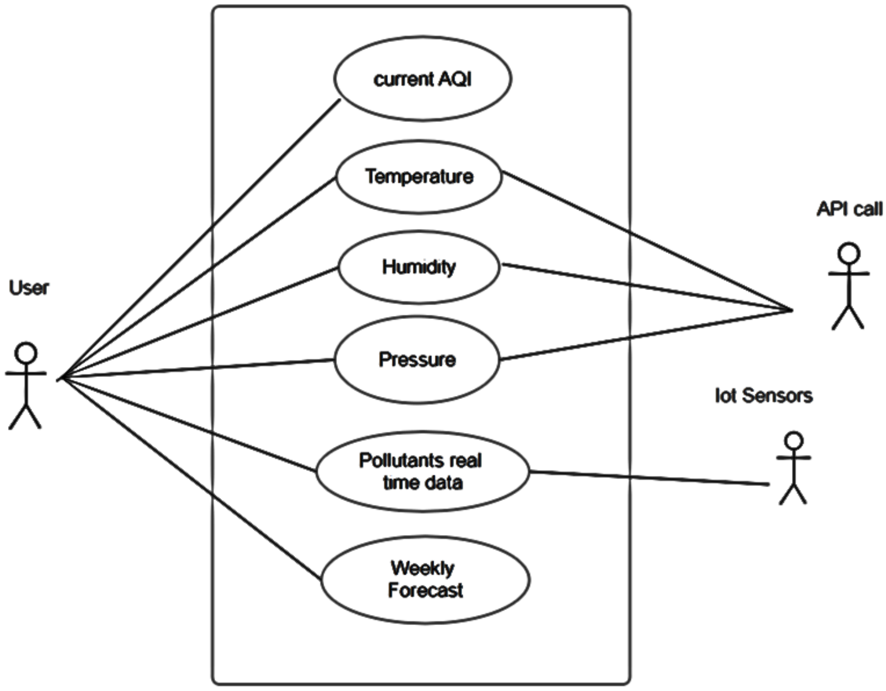

Fig 3 Use Case Diagram

4.2 Class Diagram

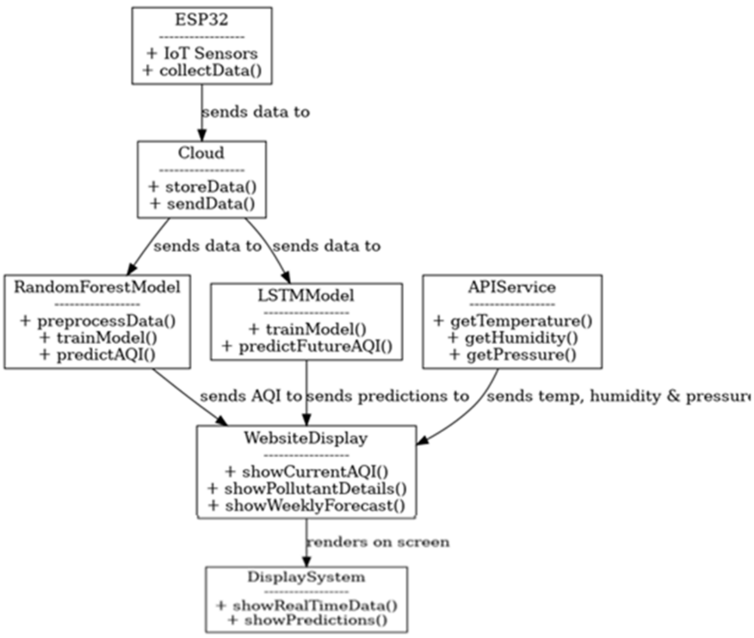

Fig 4 Class Diagram

4.3 Sequence Diagram

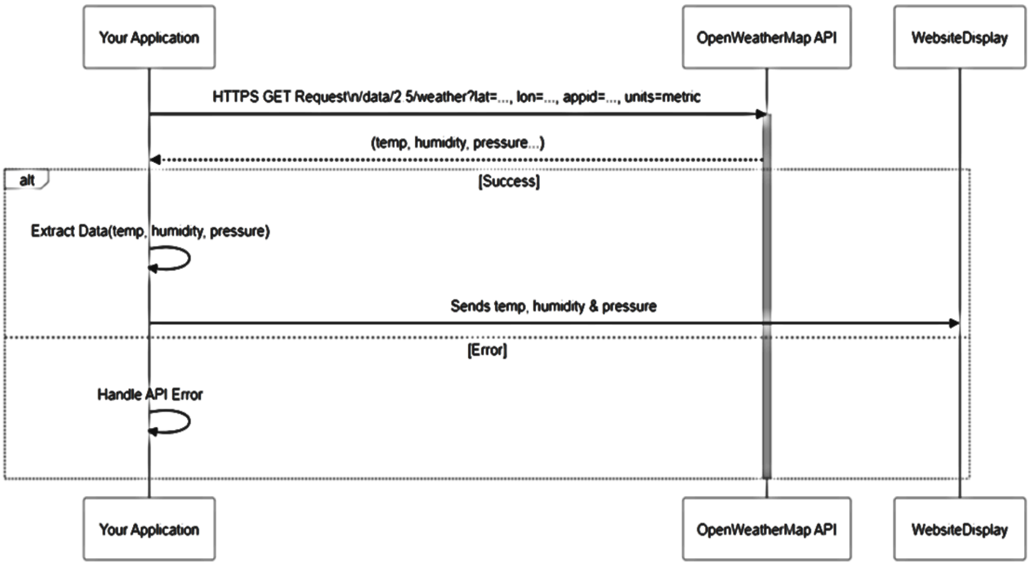

Fig 5 Sequence Diagram

4.4 Deployment Diagram

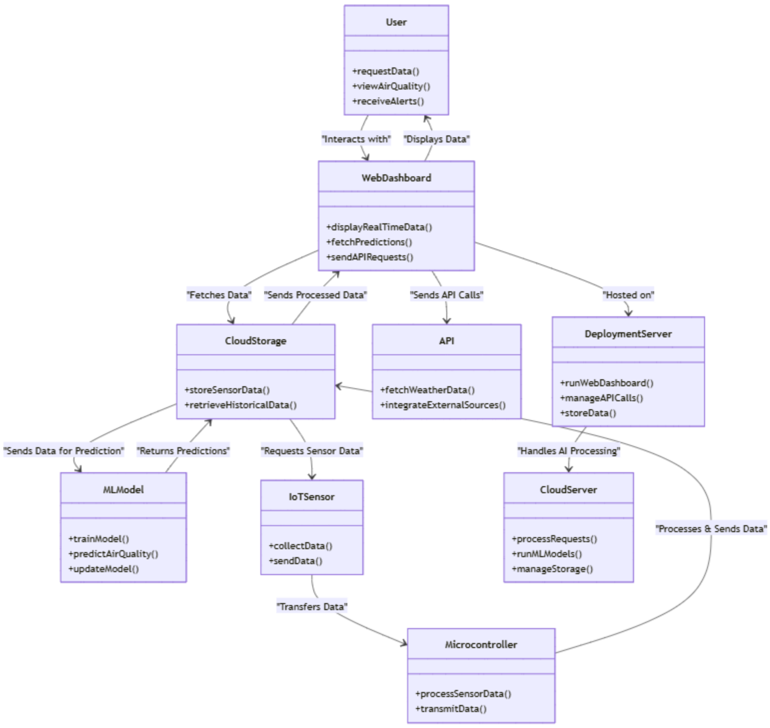

Fig 6 Deployment Diagram

4.5 State Chart Diagram

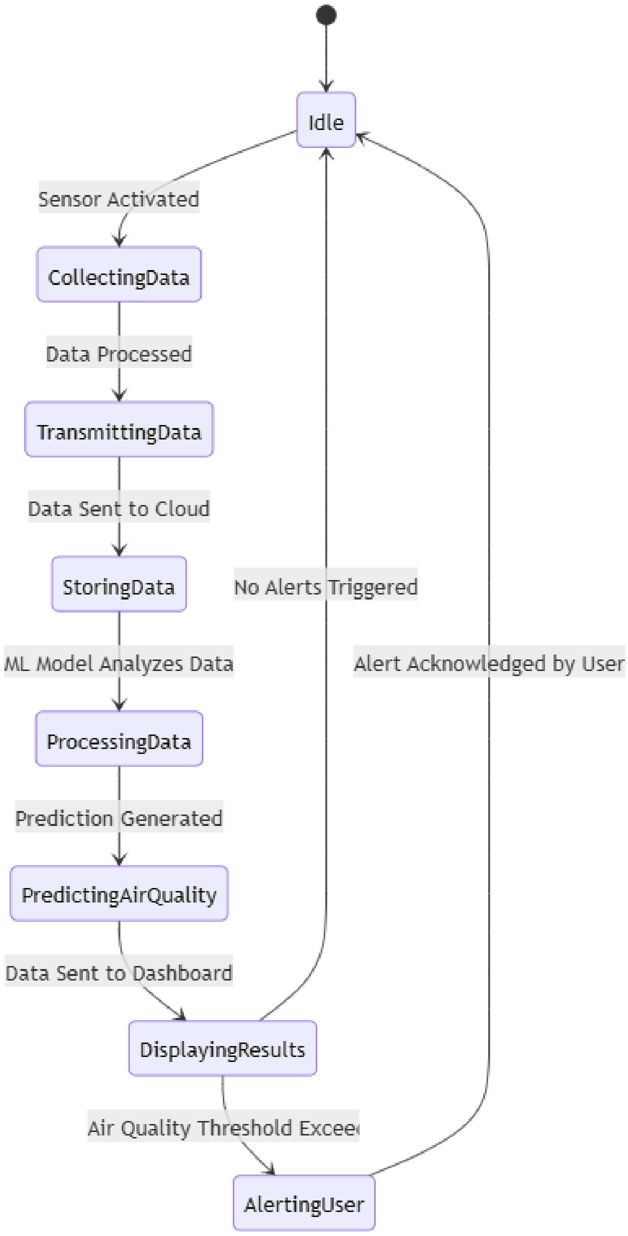

Fig 7 State Chart Diagram

4.6 Interaction Diagram

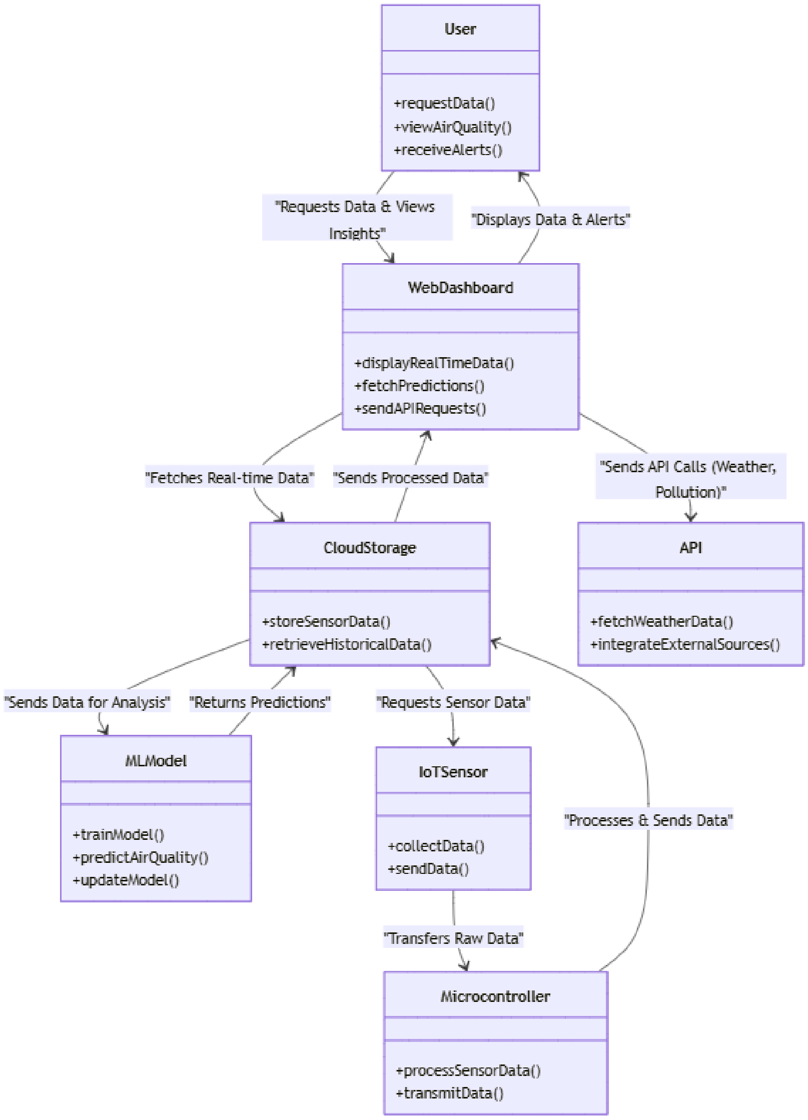

Fig 8 Interaction Diagram

Chapter 5
ARCHITECTURE

5. ARCHITECTURE

5.1 Flow Chart

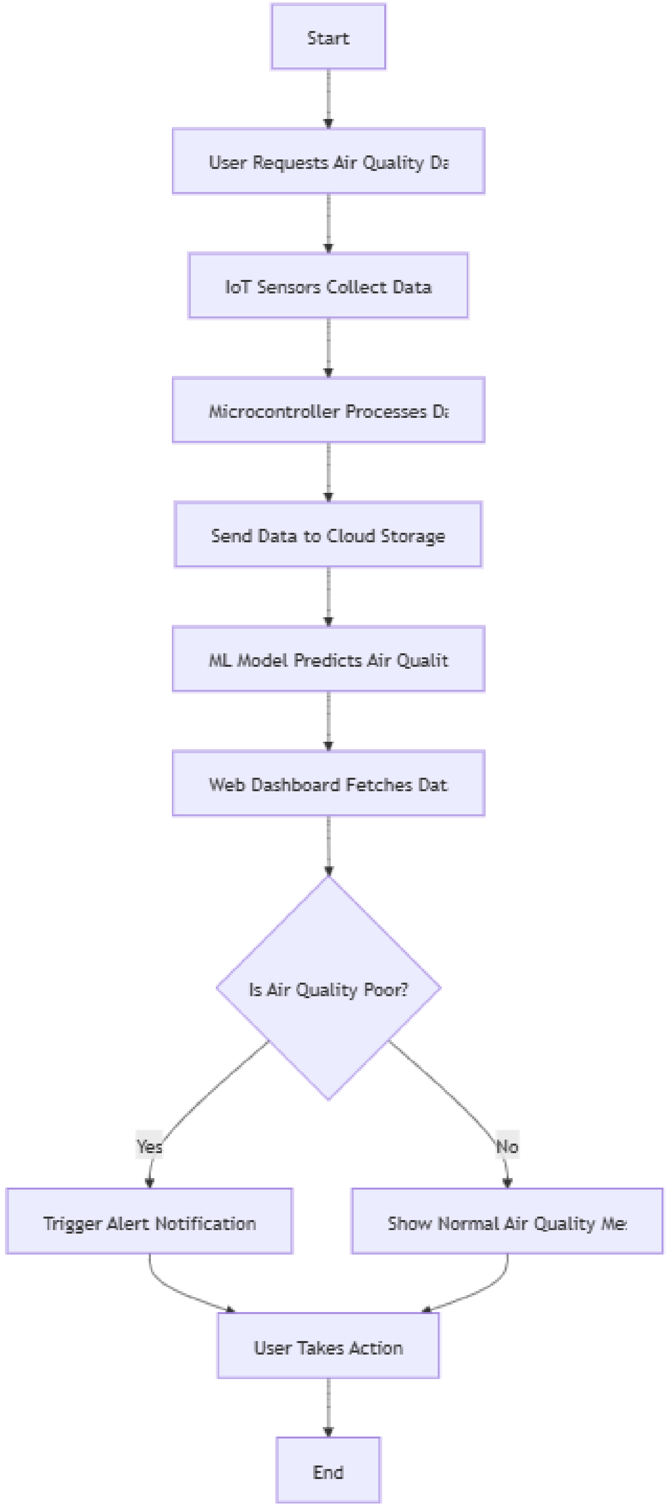

Figure 9: Flowchart

5.2 Data Collection

PM2.5	PM10	CO	NOx	NH3	Benzene	AQI
73.44	135.12	0.95	16.6	6.47	6.67	138
83.96	151.62	1	17.59	6.62	7.02	158
80.65	133.85	0.88	17.78	6.95	6.95	187
56.2	108.33	0.61	12.92	6.42	4.8	122
51.84	109.52	0.62	14.22	6.49	4.92	110
53.9	111.12	0.59	11.43	6.04	4.51	105
53.79	117.94	0.54	11.97	6.14	4.71	111
47.72	106.01	0.54	10.48	6.15	4.48	108
46.9	108.1	0.73	17.37	6.75	5.49	104
71.66	130.66	0.91	21.56	7.01	6.21	110
92.3	159.18	0.99	17.86	6.61	3.33	190
104.11	150.29	0.82	13.52	6.22	3	231
122.16	180.25	0.73	11.84	7.79	3.04	253
154.85	224.22	0.82	33.08	27.94	3.21	335
110.06	184.04	1.29	43.39	40.84	3.47	268
108.09	181.54	1.01	20.99	22.6	3.42	270
109.17	169.19	0.99	13.98	12.73	3.29	262
112.71	190.12	0.89	14.1	12.82	3.43	270
123.26	190.28	1.02	18.63	19.21	4.25	297
135.98	215.41	0.98	5.18	5.93	5.01	283
114.56	166.58	0.69	7.82	4.2	4.44	322
53.96	115.54	0.54	9.39	6.12	4.09	148
58.85	125.5	1.02	13.85	17.58	3.88	117
46.3	84.99	0.79	14.7	17.99	3.67	100
45.3	72.48	0.74	35.9	7.29	3.36	82

5.3 Module 1

User Request Module

The process starts when the user requests real-time air quality data through an application, such as a mobile app or web-based interface. This request serves as a trigger for the system to begin data collection. The interface is designed to be user-friendly, allowing users to check air quality in different locations with minimal interaction.

5.4 Module 2

Data Collection Module

Once the request is made, IoT sensors deployed in the environment begin collecting air quality parameters. These sensors measure pollutants such as **PM2.5, PM10, CO2, CO, SO2, and NO2**, as well as temperature and humidity levels. The collected data is in raw form and may contain noise or fluctuations due to environmental factors.

5.5 Module 3

Data Processing Module

The microcontroller acts as the primary processing unit, receiving raw sensor data and performing essential operations such as noise filtration, data calibration, and error correction. This step ensures that the data is structured correctly before being transmitted. The microcontroller then forwards the processed data to cloud storage, ensuring scalability and remote access.

5.6 Module 4

Cloud and Machine Learning Module

The processed data is stored securely in the cloud, where it can be accessed for further analysis. A **Machine Learning (ML) model** is applied to predict air quality trends based on historical data. The model classifies the **Air Quality Index (AQI)** into categories such as **Good, Moderate, Poor, and Hazardous.** The predictive analysis helps identify pollution patterns and warn users about potential risks in advance.

5.7 Module 5

Dashboard and Decision Module

The web-based dashboard acts as the central interface for users to view air quality reports. It fetches the analyzed data from the cloud and visually represents it using graphs, color-coded AQI levels, and historical trends. A decision-making system determines whether the air quality is within safe limits. If the air quality is acceptable, a normal message is displayed; otherwise, further actions are triggered.

5.8 Module 6

Alert and Response Module

If poor air quality is detected, the system generates an **alert notification** via email, SMS, or app notification. This alert informs users about potential health risks, advising them to take necessary precautions such as **wearing masks, staying indoors, or using air purifiers.** The user is then expected to acknowledge the alert and take appropriate action. Once this is done, the process concludes until the next monitoring cycle begins.

Chapter 6
SAMPLE CODE

6. SAMPLE CODE

6.1 Front End

```
from flask import Flask, jsonify, request, render_template
from flask_cors import CORS
import random, datetime
app = Flask(__name__)
CORS(app)
# Generate random sensor and AQI data
def generate_data():
return {
'temperature': round(random.uniform(20, 30), 1),
'humidity': round(random.uniform(40, 60), 1),
'pressure': round(random.uniform(1000, 1010), 1),
'co2': round(random.uniform(500, 600), 1),
'pm25': round(random.uniform(20, 30), 1),
'pm10': round(random.uniform(30, 45), 1),
'co': round(random.uniform(1, 2), 1),
'no2': round(random.uniform(0.05, 0.15), 1),
'aqi': round(random.uniform(50, 150), 1),
'confidence': round(random.uniform(90, 99), 1)
}
def aqi_category(aqi):
return ["Good", "Moderate", "Poor", "Very Poor", "Hazardous"][(aqi // 50) if aqi <= 200 else 4]
@app.route('/')
def index():
return render_template('index.html')
@app.route('/sensor_data')
def sensor_data():
return jsonify(generate_data())
@app.route('/forecast')
def forecast():
```

```
today = datetime.date.today()
return jsonify([{'date': (today + datetime.timedelta(days=i)).strftime('%a, %b %d'),
'aqi': (aqi := round(random.uniform(50, 200), 0)), 'category': aqi_category(aqi)} for i in range(7)])
@app.route('/hourly_forecast')
def hourly_forecast():
return jsonify([{'time': f"{h:02d}:00", 'aqi': round(random.uniform(50, 200), 0)} for h in range(24)])
if __name__ == '__main__':
app.run(debug=True, host='0.0.0.0', port=5000)
```

6.2 Back End

6.2.1 Backend Setup

```
from flask import Flask, request, jsonify, render_template
from flask_cors import CORS
import joblib
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from sklearn.preprocessing import MinMaxScaler
import os
import smtplib
from email.mime.text import MIMEText
import ssl
import time
import matplotlib.pyplot as plt
import google.generativeai as genai  # Gemini API
app = Flask(__name__)
CORS(app)  # Enable Cross-Origin Requests
```

6.2.2 Model Loading

```
# Load trained Random Forest model
try:
rf_model = joblib.load("aqi_model.pkl")
```

```
print("✅ Random Forest model loaded successfully!")
except Exception as e:
rf_model = None
print(f"❌ Error loading Random Forest model: {e}")
# Load trained LSTM model
try:
lstm_model = keras.models.load_model("aqi_lstm_model.h5", custom_objects={"mse":
keras.losses.MeanSquaredError()}, safe_mode=False)
print("✅ LSTM model loaded successfully!")
except Exception as e:
lstm_model = None
print(f"❌ Error loading LSTM model: {e}")
```

6.2.3 Email Alert System

```
# Email Configuration
SENDER_EMAIL = "your-email@gmail.com"
SENDER_PASSWORD = "your-app-password"
RECEIVER_EMAIL = "recipient-email@gmail.com"
last_email_time = 0
EMAIL_INTERVAL = 30 * 60  # 30 minutes
def send_email(aqi_value):
subject = "Air Quality Alert: AQI Exceeds Threshold!"
body = f"The Air Quality Index (AQI) is {aqi_value}, exceeding the threshold. Please take
precautions."
msg = MIMEText(body)
msg["Subject"] = subject
msg["From"] = SENDER_EMAIL
msg["To"] = RECEIVER_EMAIL
try:
context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
server.login(SENDER_EMAIL, SENDER_PASSWORD)
server.sendmail(SENDER_EMAIL, RECEIVER_EMAIL, msg.as_string())
```

```
print("📧 Email notification sent successfully!")
except Exception as e:
print(f"❌ Error sending email: {e}")
```

6.2.4 Real-Time AQI Prediction (Random Forest Model)

```
@app.route("/sensor-data", methods=["POST"])
def predict_aqi():
global last_email_time
if rf_model is None:
return jsonify({"error": "Random Forest model not loaded"}), 500
try:
data = request.get_json()
input_features = np.array([[data.get("PM2.5", 0), data.get("PM10", 0), data.get("NOx", 0),
data.get("Benzene", 0), data.get("NH3", 0), data.get("CO", 0)]])
predicted_aqi = round(rf_model.predict(input_features)[0], 2)

# Send alert if AQI exceeds 100
if predicted_aqi > 100 and (time.time() - last_email_time >= EMAIL_INTERVAL):
send_email(predicted_aqi)
return jsonify({"Predicted AQI": predicted_aqi})
except Exception as e:
return jsonify({"error": str(e)}), 500
```

6.2.5 7-Day AQI Forecast (LSTM Model)

```
@app.route("/predict-7day", methods=["GET"])
def predict_7day_aqi():
if lstm_model is None:
return jsonify({"error": "LSTM model not loaded"}), 500
try:
df = pd.read_csv("aqi_test_data.csv")
df["Date"] = pd.to_datetime(df["Date"])
df_scaled = MinMaxScaler().fit_transform(df[["PM2.5", "PM10", "NOx", "NH3", "CO",
```

```
"Benzene"]])
input_data = df_scaled[-1].reshape(1, 1, df_scaled.shape[1])
predicted_scaled = lstm_model.predict(input_data)
predicted_aqi = MinMaxScaler().fit(df[["PM2.5"]]).inverse_transform(predicted_scaled.T).flatten()

future_dates = [df["Date"].max() + pd.Timedelta(days=i) for i in range(1, 8)]
return jsonify([{"Date": str(date), "Predicted_AQI": round(aqi, 2)} for date, aqi in zip(future_dates,
predicted_aqi)])

except Exception as e:
return jsonify({"error": str(e)}), 500
```

6.2.6 Retrieve AQI Test Data

```
@app.route("/aqi-test-data")
def get_aqi_test_data():
try:
df = pd.read_csv("aqi_test_data.csv")
df["Date"] = pd.to_datetime(df["Date"])
return jsonify({
"dates": df["Date"].dt.strftime("%Y-%m-%d").tolist(),
"pm25": df["PM2.5"].tolist(),
"pm10": df["PM10"].tolist(),
"nox": df["NOx"].tolist(),
"nh3": df["NH3"].tolist(),
"co": df["CO"].tolist(),
"benzene": df["Benzene"].tolist()
})
except Exception as e:
return jsonify({"error": str(e)}), 500
```

6.2.7 Gemini AI Integration

```
@app.route("/get-gemini-response", methods=["POST"])
def get_gemini_response():
```

```
if model is None:
return jsonify({"error": "Gemini API not initialized"}), 500
try:
data = request.get_json()
aqi_value = data.get("aqi", None)
if aqi_value is None:
return jsonify({"error": "AQI value missing"}), 400
prompt = f"The AQI is {aqi_value}. Provide health precautions and safety measures."
response = model.generate_content(prompt)
return jsonify({"response": response.text})
except Exception as e:
return jsonify({"error": str(e)}), 500
```

6.2.8 Web Interface Endpoint

```
@app.route("/")
def index():
return render_template("index.html")
```

Chapter 7
RESULT AND ANALYSIS

7. RESULT AND ANALYSIS

7.1 Screen Shots

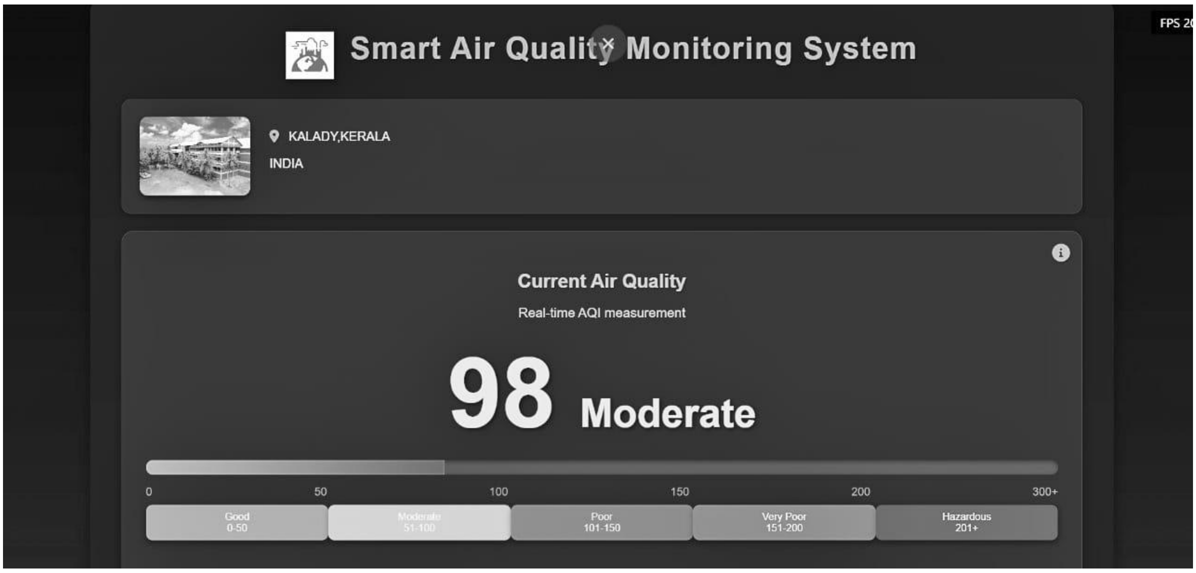

Figure 10: Home page (Current AQI monitoring)

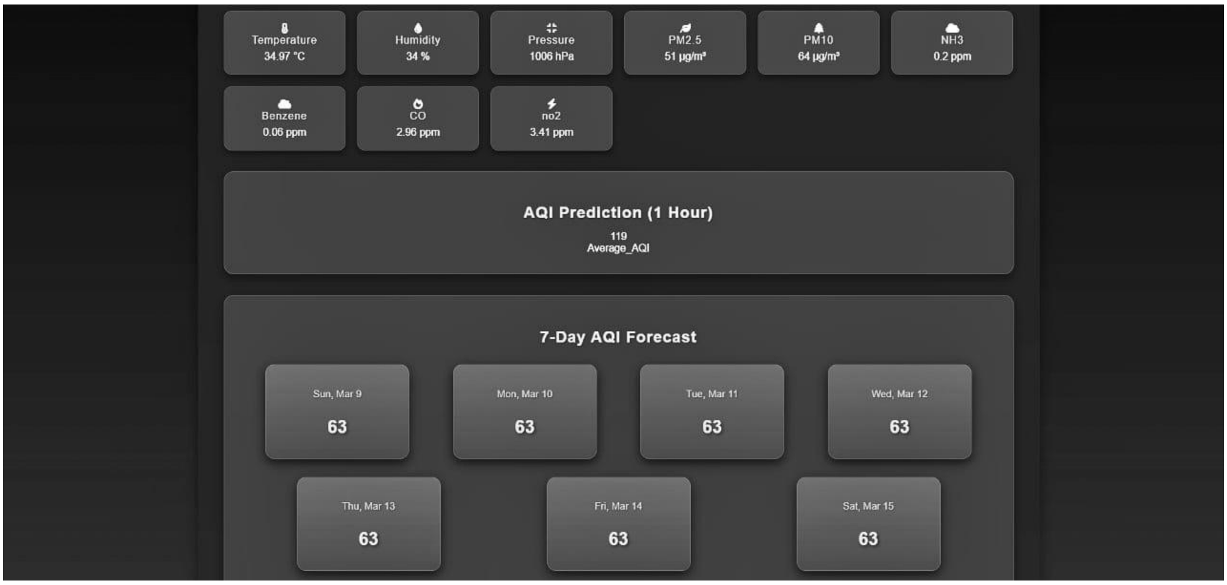

Figure 11 : Home Page 2(7-Day AQI Forecasting)

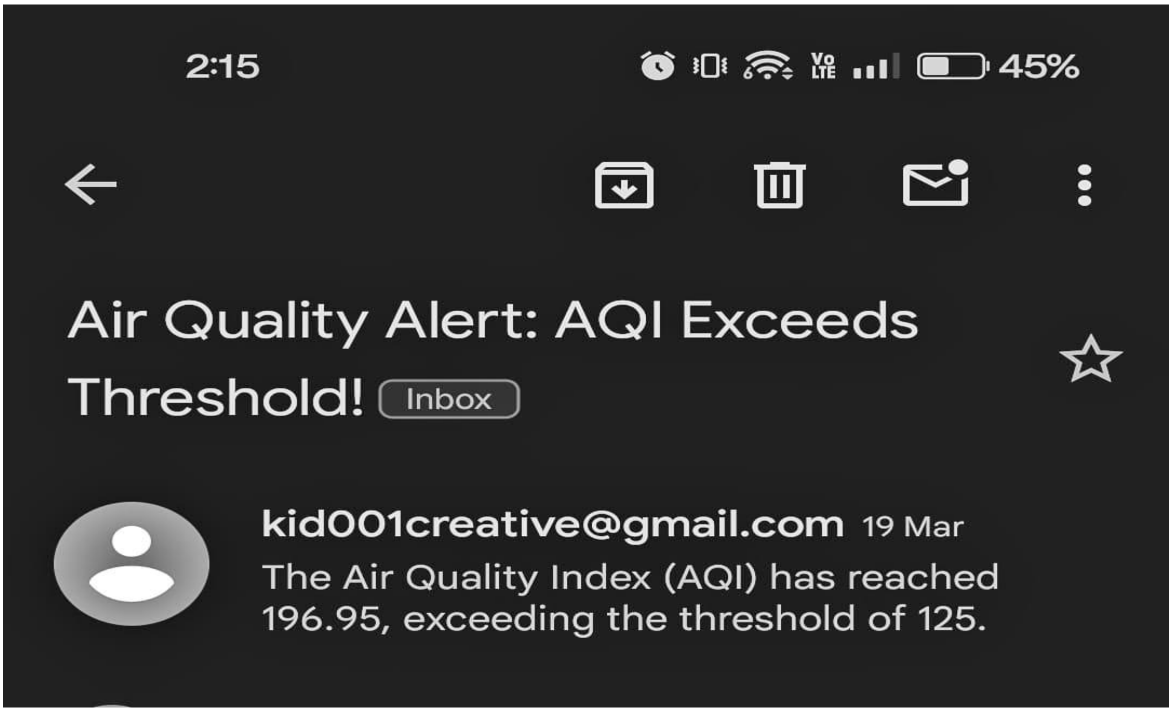

Figure 12 : Email Alert Response to User

7.2 Analysis

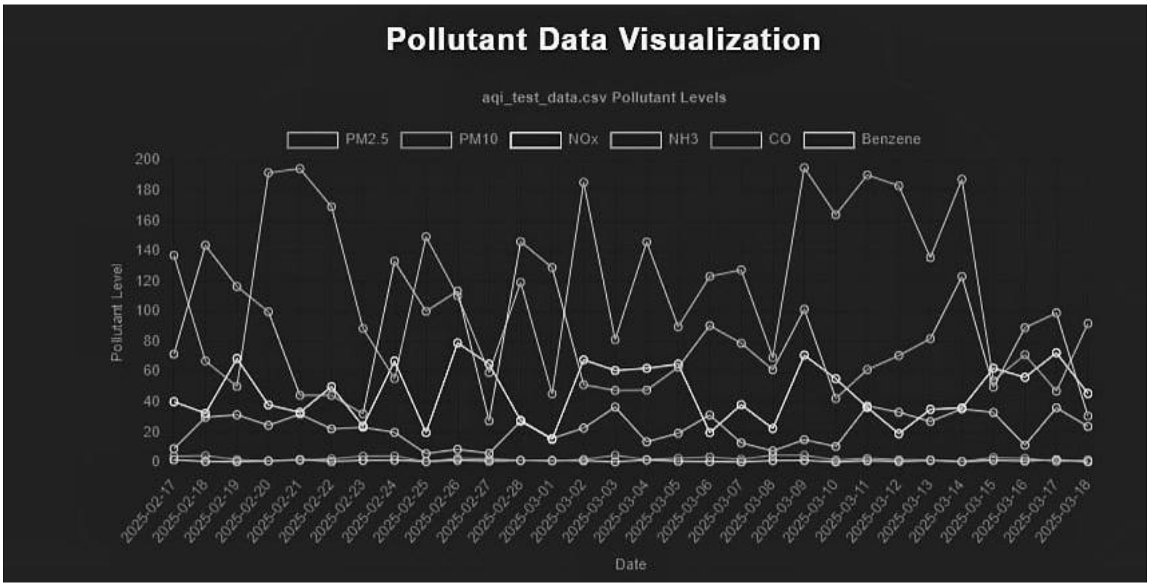

Figure 13: Data Visualization(30 Day Data Analysis of each Pollutant)

Chapter 8
CONCLUSION

8. CONCLUSION

The IoT-based air quality monitoring and prediction system effectively overcomes the limitations of traditional air quality monitoring by offering real-time pollutant tracking, accurate forecasting, and automated alerts. Unlike conventional fixed monitoring stations, which are often expensive and have limited coverage, this system provides continuous, scalable, and cost-effective air quality monitoring through a network of IoT sensors. These sensors collect data on key pollutants such as PM2.5, PM10, NO_2, CO, and NH_3, along with environmental parameters like temperature, humidity, and atmospheric pressure. The collected data is transmitted to a cloud-based storage system, ensuring secure and remote access for analysis and visualization.

A major strength of the system lies in its use of machine learning algorithms to analyze historical data and detect patterns in pollution levels. By employing predictive analytics techniques such as LSTM neural networks and Random Forest models, the system can accurately forecast air quality trends for upcoming days. This predictive capability allows for proactive pollution management, enabling policymakers, environmental agencies, and individuals to take preventive measures before air quality deteriorates to hazardous levels.

The web-based dashboard serves as an intuitive interface for users, offering real-time insights, historical data visualization, and future AQI predictions. This ensures that users, including researchers, government officials, and the general public, can easily monitor air pollution levels in different locations. Additionally, the system's automated email alert mechanism enhances responsiveness by notifying users whenever air quality drops below safe levels, allowing them to take immediate action to reduce exposure.

One of the key advantages of this project is its cost-effectiveness, scalability, and ease of deployment, making it an ideal solution for a wide range of environments, including urban centers, industrial areas, residential communities, and remote locations. Its modular design allows for easy expansion, meaning additional sensors can be deployed in different locations to provide wider coverage and improved accuracy.

By integrating IoT, cloud computing, and AI-driven predictive analytics, this system makes a significant contribution to sustainable urban development and public health. It enables smart city applications, supports data-driven policy decisions, and empowers individuals with real-time information to minimize pollution exposure. Ultimately, this project represents a step forward in proactive air quality management, promoting healthier communities and a cleaner environment for future generations.

Chapter 9
FUTURE ENHANCEMENTS

9. FUTURE ENHANCEMENT

To ensure the Air Quality Prediction and Data Analysis System remains cutting-edge and continues to provide increasing value, several future enhancements can be implemented. These include improving prediction capabilities with advanced ML models, long-term forecasting, and source apportionment, as well as expanding the sensor network and integrating additional data sources like traffic patterns and satellite imagery. The user experience can be enhanced through interactive data visualization, personalized alerts, and mobile app improvements. Furthermore, system scalability and robustness can be addressed through scalable cloud infrastructure and fault tolerance. Additional features such as indoor air quality monitoring, location-specific forecasting, and integration with smart city platforms can further extend the system's functionality and applicability.

Chapter 10
REFERENCES

10. REFERENCES

1. Rawal, R. (2019). Air Quality Monitoring System Using IoT Sensors and Cloud Platforms. International Journal of Environmental Technology.
2. Patel, D., Kulwant, M., Shirin, S., Kumar, A., Ansari, M. A., & Yadav, A. K. (2022). Artificial Intelligence for Air Quality and Control Systems. IEEE Transactions on Smart Cities.
3. Liu, Z., Chen, K., Ning, Z., Wang, L., & Zheng, Z. (2023). Research on Air Quality Prediction Based on Correlation Analysis and XGBoost. Environmental Science & Technology.
4. Dubey, K., Dubey, R., Pandey, S., Sinha, A., & Rahhore, Y. (2024). Revolutionizing Air Quality Forecasting with ML & Optimization. Journal of Applied AI in Environmental Science.
5. Banciu, C., Florea, A., & Bogdan, R. (2024). Monitoring and Predicting Air Quality with IoT Devices and Machine Learning. Sensors and Actuators B: Chemical.
6. Kaggle. (2023). Air Quality Data Sets for Machine Learning Analysis. https://www.kaggle.com/datasets
7. Environmental Protection Agency (EPA). (2023). Air Quality Trends and Forecasting Methods. https://www.epa.gov/outdoor-air-quality-data
8. OpenAQ. (2024). Global Air Quality Data for Research and Policy Implementation. https://openaq.org/
9. ThingSpeak. (2024). IoT Cloud Computing for Real-Time Environmental Monitoring. https://thingspeak.com/
10. Google Firebase. (2024). Real-Time Data Management and Cloud Integration for IoT Applications. https://firebase.google.com/docs
11. Scikit-learn Developers. (2023). Machine Learning Algorithms for Environmental Data Analysis. https://scikit-learn.org/stable/
12. TensorFlow Developers. (2024). Deep Learning Models for Time-Series Forecasting of Air Quality. https://www.tensorflow.org/ Flask Web Framework. (2024).
13. Developing Interactive Dashboards for IoT Applications. https://flask.palletsprojects.com/
14. World Health Organization (WHO). (2023). Health Effects of Air Pollution: Guidelines and Recommendations. https://www.who.int/
15. Internet of Things (IoT) Standards. (2024). IEEE Technical Standards and Guidelines for IoT-Based Monitoring Systems. https://standards.ieee.org/