AIR POLLUTION MONITORING AND PREDICTION USING IoT AND MACHINE LEARNING

Project report submitted
in partial fulfillment of the requirement for the degree of

Bachelor of Technology
In
ELECTRONICS & TELECOMMUNICATION ENGINEERING

By

Abhiraj Paul (200610026003)
Amandeep Vasistha (200610026004)
Kalpeswar Paul (200610026025)
Subhadeep Karmakar (200610026055)

Under the guidance
of
Dr. NAVAJIT SAIKIA, Associate Professor
Dr. SIDDHANTA BORAH, Assistant Professor

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING
ASSAM ENGINEERING COLLEGE
JALUKBARI- 781013, Guwahati
June, 2024

ASSAM ENGINEERING COLLEGE, GUWAHATI

CERTIFICATE

This is to certify that the project entitled "Air Pollution Monitoring and Prediction using IoT and Machine Learning" submitted by Amandeep Vasistha (200610026004), Kalpeswar Paul (200610026025), Subhadeep Karmakar (200610026055), Abhiraj Paul (200610026003) in the partial fulfillment of the requirements for the award of Bachelor of Technology degree in Electronics & Telecommunication at Assam Engineering College, Jalukbari, Guwahati is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in this report has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Signature of Supervisor

Dr. Navajit Saikia
Associate Professor (HOD)

Signature of Supervisor

Dr. Siddhanta Borah
Assistant Professor

DECLARATION

We, Amandeep Vasistha(20/348), Kalpeswar Paul(20/173), Subhadeep Karmakar(20/357), and Abhiraj Paul(20/329) declare that this written submission represents our ideas in our own words and where others' ideas or words have not been included. We have adequately cited and referenced the sources. The project has been completed by us with utmost honesty and due diligence and have not resorted to any sort of plagiarism. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented, fabricated, or falsified any idea/data/fact/source in our submission of the project and the report. We understand that any violation of the above will cause disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Roll No:	**Name:**	**Signature:**
200610026003	Abhiraj Paul	
200610026004	Amandeep Vasistha	
200610026025	Kalpeswar Paul	
200610026055	Subhadeep Karmakar	

Date:_____

ACKNOWLEDGEMENT

We take this opportunity to thank **Dr. Kalyan Kalita**, Principal, Assam Engineering College, Guwahati, our mentor **Dr. Navajit Saikia**, Head of Department, Electronics and Telecommunication Engineering, Assam Engineering College for allowing us to take up this project. Along with our HOD Sir, we also express our special thanks and gratitude to our co-mentor for this project **Dr. Siddhanta Borah**, Assistant Professor, Department of Electronics and Telecommunication Engineering, Assam Engineering College, Guwahati for his valuable guidance and supervision during the preparation of the project from time to time. We extend all our sincere thanks and gratitude to one and all of the AEC family for their cooperation in the completion of this project.

Abhiraj Paul (200610026003)

Amandeep Vasistha (200610026004)

Kalpeswar Paul (200610026025)

Subhadeep Karmakar (200610026055)

ABSTRACT

Air Pollution refers to the existence of harmful substances in the atmosphere. One popular measure for airpollution is the Air Quality Index (AQI). It rates air pollution based on pollutants present in the atmosphere. Industrialization, deforestation etc. has resulted in a steep rise in air pollution. Prolonged exposure to such conditions leads to pulmonary diseases. Thus, there is a need for a system that can monitor Air pollution. It can be used in modern cities to remotely monitor air quality levels, which will help the government take suitable measures to reduce air pollution levels. This project focuses on developing a system that combines the power of the Internet of Things and Machine Learning to monitor and predict air quality in real-time.

Along with Arduino and NodeMCU, this project comprises sensors such as MQ135, PM2.5, and MQ-7. These sensors collect the data from the atmosphere, and Arduino is used to read these data from the sensors. After the data is collected, it is transferred to the cloud using NodeMCU. Machine Learning models are built and trained by applying the algorithms on past years data trends as well as newly collected and updated datasets. These models can be used for predicting the AQI.

Random Forest Regressor has been found to be the most accurate regressor model while decision tree classifier has been found to be the most accurate classification model. Further accuracies can be obtained upon providing more accurate and updated dataset. The Arduino UNO plays a crucial role in reading and processing data from these analog sensors.

The project represents a noteworthy advancement in tackling the intricate problems caused by air pollution in urban environments. By integrating state-of-the-art technologies such as the Internet of Things (IoT) and advanced Machine Learning (ML) algorithms, the project has laid the foundation for a comprehensive and dynamic system capable of providing real time air quality insights and predictive analytics.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Introduction:

Air Pollution refers to the existence of harmful substances in the atmosphere. The Air Quality Index (AQI) is a popular measure for air pollution. It rates air pollution based on pollutants present in the atmosphere. The quality of air we breathe is a fundamental determinant of public health and environmental well-being. With urbanization and industrialization on the rise, the challenge of monitoring and managing air quality has become increasingly critical. This project addresses the imperative need for effective Air Quality Monitoring and Prediction, combining the capabilities of Internet of Things (IoT) devices and Machine Learning (ML) techniques. The convergence of these technologies offers a holistic solution to tackle the complexities associated with understanding and forecasting air pollution.

Fig 1.1 AQI chart (Source: Research Gate.net)

Urban air pollution, comprising particulate matter, nitrogen dioxide, carbon monoxide etc poses significant risks to human health, ranging from respiratory issues to more severe long-term effects. Traditional monitoring methods often lack the spatial and temporal resolution required

for real-time decision-making. This project aims to overcome these limitations by deploying a network of IoT-based sensors strategically positioned across key locations to continuously measure and relay air quality parameters.

1.2 Present day scenario:

As urbanization accelerates and industrial activities expand, the global concern for deteriorating air quality has become a pressing issue. In today's scenario, densely populated urban centers are grappling with the adverse effects of air pollution, posing significant challenges to public health and the environment. Traditional air quality monitoring systems often fall short in providing real-time, granular data necessary for timely interventions.

Contemporary efforts to monitor air quality involve a combination of stationary monitoring stations and satellite-based observations. While these methods offer valuable insights, they are limited in spatial coverage and may not capture localized variations in pollution levels. Moreover, the growing need for accurate, real-time data to address dynamic environmental conditions underscores the inadequacy of conventional monitoring approaches.

1.3 Importance of the Project:

In light of the present-day scenario, the project addresses the need for a sophisticated and integrated approach. By leveraging the capabilities of IoT for real-time data collection and ML for predictive analytics, the project aims to contribute to a paradigm shift in air quality management, providing stakeholders with the tools needed to make informed decisions and proactively address the challenges posed by air pollution.

1.4 Motivation to do the Project:

The motivation behind undertaking the project is rooted in the critical importance of addressing the escalating challenges posed by air pollution in contemporary urban environments.

Several key factors drive the motivation for this project:

1.4.1 Public Health Impact:

Air pollution has emerged as a significant threat to public health, contributing to respiratory diseases, cardiovascular issues, and other health complications. The project is motivated by a deep concern for the well-being of individuals exposed to suboptimal air quality and aims to develop a solution that empowers communities with actionable information to protect their health.

1.4.2 Limitations of Current Monitoring Systems:

Traditional air quality monitoring systems often fall short in providing real-time and localized data. The motivation arises from the recognition that contemporary challenges demand more dynamic and responsive solutions. The integration of IoT sensors addresses these limitations by offering a network that captures real-time variations across diverse locations.

1.4.3 Advancements in Technology:

The rapid advancements in IoT and Machine Learning technologies provide an opportune moment to create innovative solutions for complex challenges. The motivation stems from the desire to harness these technological capabilities to revolutionize how we monitor and manage air quality, making it more effective, accessible, and predictive.

1.4.4 Smart City Development:

As cities strive to become smarter and more sustainable, the project aligns with the vision of smart city development. The integration of IoT and Machine Learning contributes to creating intelligent urban infrastructures that can adapt to environmental changes, enhance the quality of life, and promote well-being.

1.5 Objective of the Project:

The following objectives outline the key focus areas and desired outcomes:

- Real-Time Air Quality Monitoring:

Develop and deploy a network of IoT sensors strategically positioned in urban environments to enable real-time monitoring of critical air quality parameters, including particulate matter (PM2.5 and PM10), carbon monoxide (CO), nitrogen dioxide (NO2) etc.

- Data Collection and Integration:
  Establish a centralized data hub to collect, process, and integrate the real-time data generated by the IoT sensors. Ensure data accuracy, reliability, and integrity for subsequent analysis.

- Machine Learning Model Development:
  Employ advanced Machine Learning algorithms to develop predictive models for air quality based on historical and real-time data. Explore various ML techniques, including regression models and ensemble methods, to capture complex relationships between different air quality parameters.

- User-Friendly Interface:
  Design and implement an intuitive user interface that provides stakeholders, including citizens, environmental agencies, and decision-makers, with easy access to visualizations of real-time air quality data, historical trends, and predictive analytics. Ensure user accessibility and engagement.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction:

Air pollution monitoring and prediction using the Internet of Things (IoT) and Machine Learning (ML) is a promising approach to improve air quality. The literature review will focus on a growing body of research based on the integration of Internet of Things (IoT) devices and machine learning techniques for effective air pollution management. Numerous studies highlight the significance of real-time data collection through IoT sensors to monitor air quality parameters. Machine learning algorithms play a crucial role in analyzing and predicting air pollution levels based on historical data, meteorological factors, and other relevant variables. The review emphasizes the potential of this integrated approach to enhance the accuracy of pollution forecasts, aiding in the development of proactive measures and policies for mitigating the adverse effects of air pollution on public health and the environment.

2.2 Introduction to Project Title:

The project aims to develop a robust and efficient system that can continuously monitor air quality in real-time and predict future pollution levels. By using IoT devices, such as sensors and data collectors, and implementing advanced ML algorithms, this project seeks to revolutionize the way we understand, assess, and respond to air pollution challenges.

The objectives are:

- Real-time Monitoring: Implementing IoT sensors to collect real-time data on key air pollutants such as particulate matter (PM), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), and ozone (O3).
- Data Integration and Analysis: Aggregating data from multiple IoT devices and integrating it into a centralized system for comprehensive analysis. Employing ML algorithms to identify patterns, correlations, and trends in the collected data.
- Prediction Models: Developing predictive models using machine learning algorithms to forecast air pollution levels based on historical data, meteorological factors, and other

relevant parameters. This enables proactive measures to be taken to mitigate potential pollution spikes.

- User-Friendly Interface: Creating a user-friendly interface accessible through web or mobile applications, providing real-time air quality information, historical trends, and predictive analytics. This empowers individuals, communities, and authorities to make informed decisions regarding outdoor activities and pollution control measures.

- Alert Systems: Implementing automated alert systems to notify relevant stakeholders when air pollution levels exceed predefined thresholds. This facilitates timely responses and interventions to safeguard public health.

- Environmental Impact Assessment: Conducting an assessment of the environmental impact of air pollution based on the collected data, aiding policymakers and researchers in devising effective strategies for pollution control and urban planning.

2.3 Literature Review:

2.3.1 Recent development in the work area:

The field of air pollution monitoring and prediction is rapidly evolving, driven by advancements in technology and a growing awareness of the health risks associated with poor air quality. Here are some key recent developments:

Satellite Technology: Advances in satellite technology have allowed for more comprehensive and real-time monitoring of air pollution on a global scale. Satellites equipped with sensors can detect various pollutants, including particulate matter, nitrogen dioxide, and sulfur dioxide.

- Sensor Networks and IoT: The deployment of sensor networks and Internet of Things (IoT) devices in urban areas has increased. These sensors provide localized and real-time
- data on air quality. They are often part of smart city initiatives and contribute to more accurate and granular pollution monitoring.

- Machine Learning and Data Analytics: The integration of machine learning algorithms and data analytics has enhanced the accuracy of air pollution prediction models. These models can analyze large datasets, including historical pollution data, meteorological information, and other relevant factors to predict future pollution levels.

- Air Quality Apps and Citizen Science: There has been a rise in the development of mobile applications that provide real-time air quality information to users. Additionally, citizen science initiatives encourage the public to contribute data through personal monitoring devices, further expanding the data available for analysis.

- Government Regulations and Open Data Initiatives: Many governments are implementing stricter regulations on air quality standards. Simultaneously, there is an increasing trend towards making air quality data openly accessible to the public, researchers, and businesses. This transparency fosters innovation and collaboration in addressing air pollution issues.

- Drones for Monitoring: The use of drones equipped with air quality sensors allows for more flexible and dynamic monitoring of pollution sources, especially in areas that are challenging to access.

- Integration with Climate Models: Some efforts are being made to integrate air quality monitoring and prediction with climate models to better understand the interactions between air pollution and broader environmental changes.

2.3.2 Brief background theory:

Air pollution is the presence of harmful substances in the Earth's atmosphere as a result of human activities such as industrial processes, transportation, and the use of fossil fuels.

Particulate matter, nitrogen dioxide, sulphur dioxide, carbon monoxide, and ozone are all examples of common pollutants. These pollutants can have serious consequences for both human health and the environment. Air pollution has been linked to respiratory and cardiovascular diseases, as well as increased mortality rates. Fine particulate matter can penetrate deep into the lungs, causing

respiratory problems, whereas ground-level ozone can irritate the respiratory system. Furthermore, air pollution contributes to environmental issues such as acid rain, smog formation, and ecosystem deterioration. Adopting cleaner technologies, promoting sustainable practices, and enforcing stringent regulations to protect human health and the planet's ecosystems are all part of efforts to reduce air pollution.

Traditional methods of air quality monitoring involve the use of specialized equipment and techniques to measure the concentration of various air pollutants in the atmosphere. Common pollutants monitored include particulate matter (PM), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), ozone (O3), and volatile organic compounds (VOCs). Here are some traditional methods:

- Ambient Air Quality Monitoring Stations: These stations are strategically located in urban and industrial areas to collect air samples. They house instruments that analyze the concentration of pollutants over specific time intervals.

- Chemical Analyzers: Instruments like gas chromatographs and mass spectrometers are used to analyze air samples for specific pollutants. These tools can provide accurate measurements of various gases and particulate matter.

- Gravimetric Method: This method involves collecting particulate matter on filters and then weighing the filters before and after sampling to determine the mass concentration of particles.

- Gas Analyzers: Devices such as non-dispersive infrared (NDIR) analyzers, chemiluminescence analyzers, and UV absorption analyzers are employed to measure the concentration of gases like CO, NO2, and SO2.

- Passive Samplers: These are simple devices that absorb pollutants over a period of time. After exposure, the samplers are analyzed in a laboratory to determine the average concentration of pollutants.

- Meteorological Instruments: Understanding meteorological conditions is crucial for interpreting air quality data. Instruments such as anemometers, barometers, and thermometers are used to monitor temperature, wind speed, and atmospheric pressure.

Traditional methods of air quality monitoring, while effective, come with certain limitations. One significant limitation is the spatial coverage provided by stationary monitoring stations. These stations are often strategically located in urban and industrial areas, leaving vast regions without adequate monitoring. As a result, variations in air quality across different locations may be overlooked, hindering the development of a comprehensive understanding of the overall air quality landscape.

Furthermore, traditional methods can be resource-intensive and expensive to establish and maintain. The deployment of specialized monitoring equipment and the need for skilled personnel to operate and manage these stations contribute to high operational costs. This cost factor can limit the number of monitoring stations, again leading to gaps in data representation. Additionally, the temporal resolution of traditional methods may not capture short-term fluctuations in air quality, potentially missing crucial information about transient pollution events.

To overcome these limitations, there is a growing emphasis on using the Internet of Things (IoT) and Machine Learning (ML) technologies for air quality monitoring and prediction. The IoT allows for the deployment of a network of interconnected sensors, including low-cost and mobiledevices, across a wide geographical area. These sensors can provide real-time data, enabling a more dynamic and detailed understanding of air quality variations. ML algorithms can then processthis vast amount of data, identifying patterns and trends that may be challenging for traditional methods to discern. By integrating historical data, meteorological information, and other relevantvariables, ML models can enhance the accuracy of air quality predictions, offering valuable insights into potential pollution sources and enabling more effective mitigation strategies. The combination of IoT and ML not only addresses the limitations of traditional monitoring methods but also opens up new possibilities for proactive and data-driven approaches to air quality management.

2.3.3 Literature Survey:

Reviewing and summarising important research papers and articles on the “Integration of Internet of Things (IoT) and Machine Learning (ML) for Air Quality Monitoring and Prediction." to get an in-depth insight view of the topic.

G. Kalaivani, P. Mayilvahanan[1] paper discusses various studies on ML algorithms for AP prediction and monitoring, summarizing real-time and historical data, and discussing recent research methodologies and challenges in real-time monitoring and AQ prediction.

Ssneha Balasubramanian, T. Sneha, Saraswathi, Vinushiya B[2] paper investigates air pollution and its harmful effects on human health and the environment. The paper proposes a system utilizing sensors to monitor air quality. Additionally, the system employs machine learning models to forecast the Air Quality Index for the next five hours. The project aims to provide a remote air quality monitoring solution for major cities, contributing to efforts to reduce air pollution.

Temesegan Walelign Ayele, and Rutvik Mehta[3] paper discusses the concept of the Internet of Things (IoT), which involves interconnected computing devices, machines, objects, or people with unique identifiers capable of exchanging information without direct human involvement. The focus of the proposed work is an IoT-based system for monitoring and predicting air pollution. This system aims to analyze and forecast air quality in a specific area by combining IoT with a machine learning algorithm known as Recurrent Neural Network, specifically Long Short-Term Memory (LSTM).

T. S. Kitchilan, M. K. Abeyratne, and E. Ediriweera[4] paper discusses the increasing air pollution in urban areas, attributing it to factors such as vehicular emissions, industrial activities, and various pollutants. Recognizing the need for an efficient and cost-effective air quality monitoring system, the study introduces a device using Arduino-based sensors. After calibrating the sensors for improved accuracy, the device was used over three months to collect data, which was then employed to create forecast models for pollutant concentrations. The study highlights the successful implementation of an Internet of Things (IoT) based, real-time air quality monitoring

solution. The use of Machine Learning models showcases the potential of this approach for forecasting air quality.

Quynh Anh Tran, Quang Hung Dang, Tung Le, Huy-Tien Nguyen, and T. Le[5] paper discusses the persistent issue of air pollution and the limitations of existing studies in the field. They propose an IoT-based Air Quality Monitoring and Forecasting System that employs low-cost sensors and an Arduino UNO R3. The system utilizes machine learning algorithms, with K-Nearest-Neighbour outperforming others in terms of accuracy and execution time. Additionally, Autoregressive Integrated Moving Average and Long Short-Term Memory algorithms are applied for future air quality prediction, showing 96% accuracy for the next hour. The paper concludes with developing a real-time web interface for monitoring and forecasting air quality.

G. Gomathi, J. Jeba Emilyn, A. Sam Thamburaj, and Vinod Kumar D[6] paper discusses the significant problem of air pollution in urban areas, emphasizing the health risks associated with particulate matter (PM). To address this issue, the author proposes the use of the Internet of Things (IoT) to create a real-time air pollution monitoring and forecasting system. The system employs a deep learning approach called Long Short-Term Memory (LSTM) with ADAM optimization to estimate PM levels in the air. The LSTM model, suitable for time-series data, can predict pollution levels an hour in advance. The study highlights the advantages of using deep learning, specifically LSTM, over traditional machine learning methods. The proposed system, implemented using the Keras framework and SPYDER tool with Python, aims to enable proactive measures such as diverting traffic to alternate routes based on anticipated pollution levels.

Xinlei Liu, Wenan Tan, and Shan Tang[7] paper addresses the escalating air pollution issue in many Chinese cities, focusing on Beijing. The paper introduces a novel approach by incorporating the Gradient Boosting Decision Tree (GBDT) method into the Bagging framework, resulting in a predictive model called Bagging-GBDT. This model aims to forecast PM2.5 concentration for the next 48 hours in Beijing. To assess its effectiveness, the author compares the Bagging-GBDT model with support vector machine regression (SVR) and random forest models, using statistical indicators such as RMSE, MAE, and R2. The experimental results reveal that the Bagging-GBDT

model outperforms SVR and random forest models, demonstrating superior predictive accuracy by reducing both bias and variance.

R. Aditya C, C. R. Deshmukh, K. NayanaD, P. Gandhi, and Vidyav[8] paper discusses the significance of regulating air quality in populated and developing countries, emphasizing the role of various factors such as meteorological conditions, traffic, burning of fossil fuels, and industrial emissions, particularly focusing on Particulate Matter (PM 2.5). High levels of PM 2.5 can have serious health implications, necessitating constant monitoring and control. The paper proposes the use of Logistic regression for detecting pollution in air quality data samples and Autoregression for predicting future PM 2.5 values based on historical readings. The goal is to enable proactive measures to reduce PM 2.5 levels within safe limits by predicting future pollution levels using a dataset of daily atmospheric conditions in a specific city.

Leeban Moses, Tamilselvan, Raju, and Karthikeyan[9] paper address the global challenge of air pollution resulting from increasing urbanization and globalization, emphasizing its adverse health impacts. To combat this issue, the proposal involves implementing a cloud-based Internet of Things (IoT) system for air quality monitoring. Utilizing sensors measuring key pollutants and environmental conditions, data is transmitted to a cloud platform. A web-based application with Google Maps integration provides real-time pollution updates, empowering individuals to make informed decisions about their surroundings. The predictive analysis component, using neural networks and SVMR algorithms, aims to forecast particulate matter levels over time, enabling proactive measures for minimizing exposure to poor air quality conditions. The ultimate goal is to create a connected environment that supports healthier living and informed decision-making regarding pollution exposure.

Varsha Hable-Khandekar, and Pravin Srinath[10] paper address the global issue of air pollution and its impact on human health, highlighting the increasing attention it has garnered from researchers. The focus is on real-time air quality monitoring and forecasting, with a significant emphasis on Machine Learning (ML) as an analytical tool. The paper provides a comprehensive overview of recent advancements in air quality forecasting models and monitoring techniques, analyzing their merits and demerits, along with a comparative assessment of methodologies. The

intended audience includes those interested in understanding the current status of air quality research, past achievements, and future research questions that need to be addressed. The paper aims to contribute valuable insights for researchers, policymakers, and decision-makers involved in environmental issues.

2.4 Summarized outcome of literature review:

Air quality monitoring and prediction using the Internet of Things (IoT) and Machine Learning (ML) has received a lot of interest in the literature. The following are the summarized outcomes of the existing research:

- Key Themes and Findings: The integration of IoT and ML in air quality monitoring has gained significant attention, enabling real-time data collection and analysis. Studies emphasize the importance of sensor networks for measuring various air pollutants, such as particulate matter (PM), nitrogen dioxide (NO2), and ozone (O3). ML algorithms, particularly regression and classification models, are frequently employed to analyze and predict air quality based on collected data.

- Methodological Approaches: Various IoT devices, including sensors and actuators, are utilized for data collection and transmission in air quality monitoring networks. ML models, ranging from traditional regression models to advanced deep learning algorithms, are applied to process large datasets and predict air quality parameters. Studies often employ historical data alongside real-time sensor readings to enhance prediction accuracy.

- Consensus and Controversies: Consensus exists on the effectiveness of IoT and ML in providing accurate and timely air quality information. Controversies revolve around the selection of optimal sensor types, calibration methods, and the interpretability of complex ML models.
- Gaps in the Literature: There is very limited research that focuses on the standardization of IoT devices and data formats, hindering interoperability between different air quality monitoring systems. There is a need for studies addressing the environmental impact and sustainability of deploying extensive IoT sensor networks for air quality monitoring.

2.5 Theoretical discussions:

The theoretical foundations of the literature on air quality prediction and monitoring using the Internet of Things (IoT) and Machine Learning (ML) integration show an important development in environmental sensing techniques. Fundamentally, ubiquitous computing is supported by the integration of IoT devices into networks for monitoring air quality. According to this theoretical framework, real-time data streams from many sources converge in a ubiquitous and networked environment, which makes it easier to comprehend the dynamics of air quality. In IoT-enabled air quality monitoring, where constant, undetectable data collection fosters the development of a rich information ecosystem, the idea of ubiquitous computing is especially relevant.

Theoretical discussions frequently touch on the use of machine learning algorithms to analyze the enormous datasets produced by Internet of Things devices. With its ability to recognize patterns and provide predictive insights, machine learning fits well with the constructivist approach to research. Meaningful patterns and trends may be extracted from complicated Air quality datasets using machine learning (ML) models, which can range from simple regression approaches to more complex deep learning structures. The underlying theory in this case is based on the idea that these models may reveal latent links in the data, enabling precise forecasts and well-informed choices.

The literature review reflects a theoretical emphasis on the link between environmental science and data science. This interdisciplinary approach aligns with the systems theory, recognizing the environment as a complex, interconnected system influenced by a multitude of factors. The integration of IoT and ML technologies represents a departure from traditional reductionist approaches to air quality monitoring, embracing a holistic perspective that considers the dynamic interactions between environmental variables. In this theoretical framework, the deployment of IoT devices and ML models is not merely a technological innovation but a paradigm shift in how we conceptualize and approach the monitoring and prediction of air quality.

2.6 General Analysis:

The literature review on air quality monitoring and prediction utilizing the Internet of Things (IoT)

and Machine Learning (ML) reflects a growing body of research that addresses the critical need for effective and timely solutions to mitigate the adverse impacts of air pollution. The convergence of IoT and ML technologies has paved the way for advanced and data-driven approaches to monitor and predict air quality, opening up exciting new possibilities for better environmental management.

Numerous studies have explored the integration of IoT devices, such as sensors and monitoring stations, to collect real-time air quality data. These devices, interconnected through the IoT framework, enable a comprehensive and dynamic understanding of pollutant levels, allowing for more accurate assessments of air quality. The literature emphasizes the importance of a dense sensor network, spatially distributed to capture variations across different locations, ensuring a holistic view of air quality dynamics.

The literature acknowledges certain challenges and limitations, including the requirement for standardized protocols, data quality assurance, and the creation of models that are widely accepted. Furthermore, challenges about the expandability and energy economy of Internet of Things gadgets have been recognized as possible obstacles to their extensive integration.

2.7 Conclusion:

In conclusion, the literature review of this project underscores a transformative paradigm in environmental research and management. The synthesis of IoT technologies for real-time data acquisition and ML algorithms for predictive modeling has demonstrated great potential for addressing the complex challenges posed by air pollution. The reviewed studies collectively emphasize the importance of a comprehensive and interconnected sensor network, leveraging advanced ML techniques to enhance the precision and reliability of air quality predictions. Despite notable advancements, the literature also recognizes persistent challenges such as standardization, data quality assurance, and scalability. As the field evolves, ongoing research efforts are essential to overcoming these hurdles and realizing the full potential of IoT and ML applications in revolutionizing air quality monitoring, ultimately contributing to more effective environmental policies and public health outcomes.

CHAPTER 3

METHODOLOGY

3.1 Introduction:

This chapter discusses the methodology of the Project. It will comprise of all the components utilized as a part of this project. The role and functioning of the components will be discussed in details and further the working procedure will be discussed and along with it the working procedure will be explained as a whole with the help of a functional block diagram.

3.2 Air Pollution Measurement:

Air Quality Measurement is in general a pretty broad concept inside which mainly 2 perspectives are involved and based upon those perspectives different components and subcomponents are selected. Air Quality Measurement is classified into 2 zones such as:

- Indoor Zone:

  In this zone no prediction or forecasting is involved. Our hardware model is based upon this perspective. In this zone normal pollutant detection sensors are used such as MQ-7; MQ-135; PM 2.5. As urbanization, industrialization etc. are on steep rise in recent scenario thus it becomes difficult to maintain the optimum Air Quality level in the indoor environment. For instance, let us consider a hospital present in a particular locality and in the same locality some construction work is going on for which many trees were been cut and thus air quality gets deteriorated. Now, inside the hospital many surgeries, operations etc. are conducted for which the patients need to be kept in a controlled environment. Now as outside environment gets contaminated so in the similar manner inside room condition also gets deteriorated. So, in this situation a pollution monitoring system can be used which helps in keeping track of air quality data which is stored in .csv format which can be used for future reference. Similarly, such air pollution systems can be installed at drug manufacturing industries, Lens manufacturing industries etc. where the final stress test takes place in a controlled indoor

environment. Our Hardware model is built specifically for this purpose where it can detect the air quality in the indoor environment and we can visualize that data using a dashboard which we have made ourself.

- Outdoor Zone:

This zone comes under the jurisdiction of pollution control board specifically Central Pollution Control Board (CPCB) in India. They make use of Air Pollution monitoring system which uses sophisticated sensors specially calibrated by various scientists for different geographic zones. They monitor individual regions such as Guwahati, Kolkata etc. and integrate satellite data for efficient pollution monitoring and use traditional forecasting methods for Air Pollution measurement. For our Project we visited Pollution Control Board Assam in Bamunimaidan, Guwahati and have collected dataset for the Guwahati region. We have made a prediction system using Machine Learning involving different regression and classification models.

3.3 Methodology:

This section discusses the detailed methodologies of the project. As in this project different hardware components, software components etc. are used so for a systematic understanding, this project is mainly divided into 4 sections which are again divided into required sub-sections. The sections are:

- Functional Block diagram.
- Hardware assembly.
- Models used for prediction.
- Implementation.
- Proposed solution for integration of additional factors.

3.3.1 Functional Block-diagram:

The entire working of the project can be visualized from the block-diagram as shown in the figure below

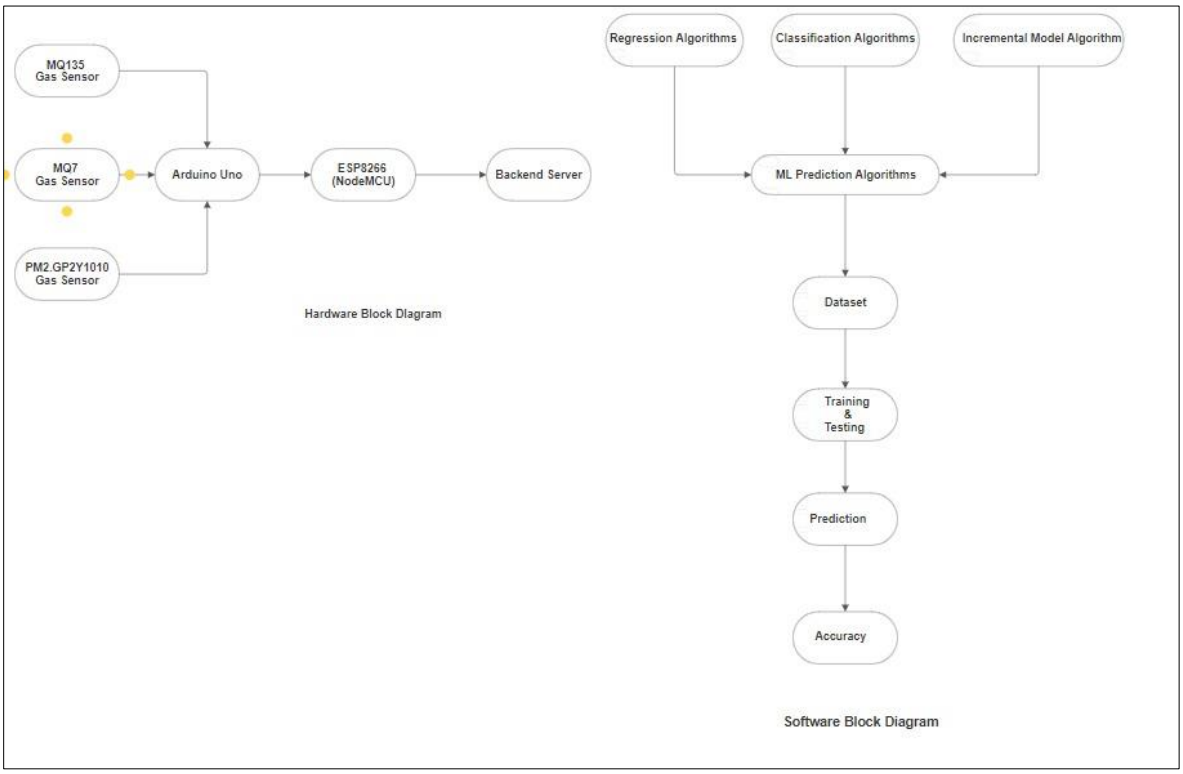

Fig 3.1: Block diagram

The block diagram is divided into two parts:

(A) Hardware section:

- This section comprises of different components such as the MQ-135 sensor, PM 2.5 sensor,Arduino UNO, Node MCU and MQ-7 sensor
- MQ135, PM2.5 and MQ-7 sensors collect data from the environment regarding the various pollutant concentration present in the atmosphere.
- The Arduino-UNO provides the regulated power supply to the 3 sensors via the VCC and GND pin and receives the sensor data via the Analog Input Pin.
- It transmits the data to the ESP8266 (NodeMCU) through the Tx-Rx port which is the serial communication port of Arduino-UNO.

- The ESP8266 (NodeMCU), when connected to a Wi-Fi Network is able to transmit the sensor data to a backend server.

(B) Software section:

- The dataset that we have selected for this project is from 2015 to 2020 for different cities of India. The dataset table consists of 737406 X 16 entries and has important information labels like PM2.5 PM10 NO NO2 NOx NH3 CO SO2 O3 Benzene AQI.
- After all the data preprocessing, the dataset is then split for the training and testing part. The training dataset is fed into different Machine-learning regression models as well as Classification Models for training the model and then the accuracy or the errors are measured using factors like RMSE, RSquared, etc.
- Then the trained model is used for predicting the AQI for the unknown value of the dataset in case of regression models while for classification models, the AQI_bucket is predicted.

3.3.2 Hardware Assembly:

1. Component specification:The components used are
    - MQ-135
    - PM 2.5 GP2Y1010AU0F
    - MQ-7
    - Arduino UNO
    - ESP8266 (NodeMCU)

   Briefly discussing each component:

   MQ-135:

   The MQ-135 is a gas sensor that detects a variety of air pollutants, including ammonia, methane, carbon dioxide, and smoke. It is commonly used for monitoring indoor air quality and can be integrated into air pollution measurement system.

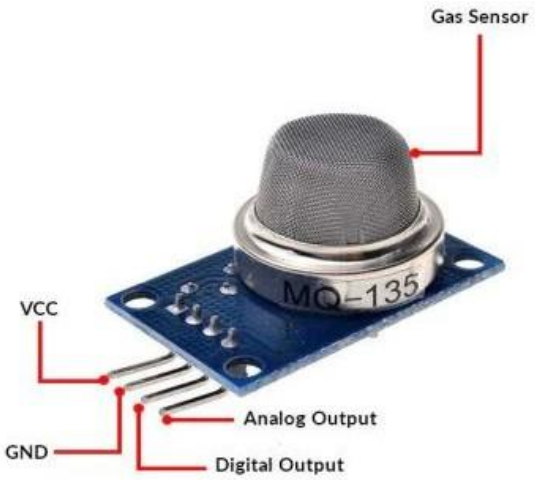

Fig 3.2: MQ-135 Sensor

PM2.5 GP2Y1010AU0F:

Particulate matter with a diameter of 2.5 micrometers or less is referred to as PM 2.5. Inhaled, this tiny particulate matter can have harmful consequences on one's health and is a major air quality indicator. PM 2.5 sensors are essential for tracking and quantifying these airborne particles.

Fig 3.3: PM2.5 GP2Y1010AU0F sensor

MQ-7:

The MQ-7 is a gas sensor designed to detect levels of carbon monoxide

(CO) in the air. It is widely used in applications where monitoring these gases is crucial, such as in homes, industries, and environmental monitoring systems.

Fig 3.4: MQ-7 Sensor

Arduino UNO:

Based on the ATmega328P microcontroller, the Arduino UNO is a popular open-source microcontroller board. It is appropriate for a variety of tasks because it has both digital &analog input/output pins.

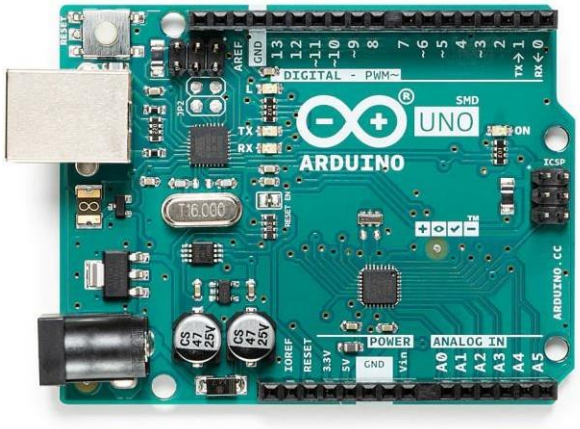

Fig 3.5: Arduino Uno

Node MCU:

It is based on the ESP8266 microcontroller. It includes firmware that runs on the ESP8266 Wi-Fi SoC. The NodeMCU functions as the receiver of the sensor data transmitted by the Arduino UNO. It transmits the sensor data to a server after it is connected to a Wi-Fi

network. The data is transmitted to the backend Server.

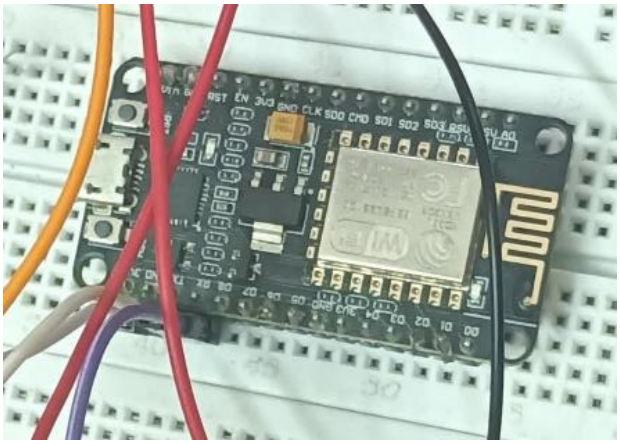

Fig 3.6: Node MCU

2.Calibration:

The sensors such as MQ-7, MQ-135, PM2.5 detect the bit value of the pollutants when they are connected as a whole all together. Since we will be calculating the AQI value for which concentration of the pollutants in PPM is needed. For this reason, mapping of the bit values of the individual sensors is done. This process as a whole is called as calibration.

Steps for calibration of MQ-135:

(i) Collect the analog bit-value using analog read function.

(ii) Convert the bit value to voltage using:

Vl=(Bit value * 5)/1023

(iii) Calculate the sensor resistance (in KiloOhm), Rs:

Rs = [(5/Vl)-1]*20

(iv) Calculate Co-efficent after selecting a base resistance:

Coefficient= Rs/R0

(v) Apply the coefficient in the PPM equation generated from the datasheet:

PPM= a*(coefficient)^b ; where a=121.451 and b=-2.780

Steps for calibration of MQ-7:

(i) Collect the analog bit-value using analog read function.

(ii) Convert the bit value to voltage using:

Vl=(Bit value * 5)/1023

(iii) Calculate the sensor resistance (in KiloOhm), Rs:

Rs = [(5/Vl)-1]*10

(iv) Calculate Co-efficent after selecting a base resistance:

Coefficient= Rs/R0.

(v) Apply the coefficient in the PPM equation generated from the datasheet:

PPM= [Coefficient/a]^(1/b) ; where a =1.0 and b = -0.45

Steps for calibration of PM2.5:

(i) Select sampling time, sleep time, delta time

(ii) Read the bit-value using analog read function.

(iii) Convert the bit value to voltage using:

Vl=(Bit value * 5)/1023

(iv) Put the Vl in the dust equation generated from the graph in the datasheet:

Density= 170*(a-1)

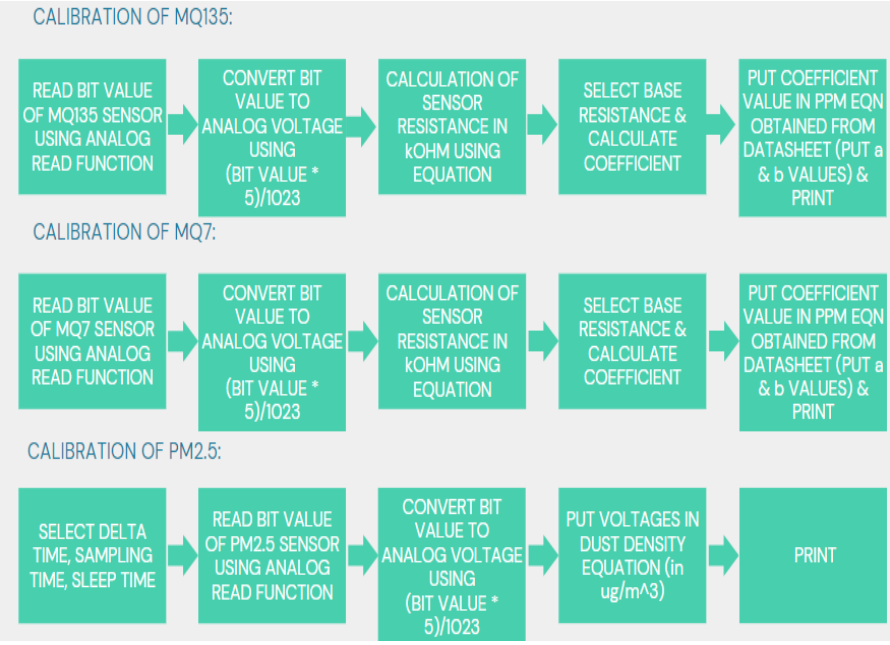

Fig 3.7: Flowchart of calibration process.

3. Circuit Diagram:

All the components such as Arduino UNO, Node MCU, MQ-7, MQ-135, PM 2.5 are connected with one another and thus the resultant layout is shown in the figure below:

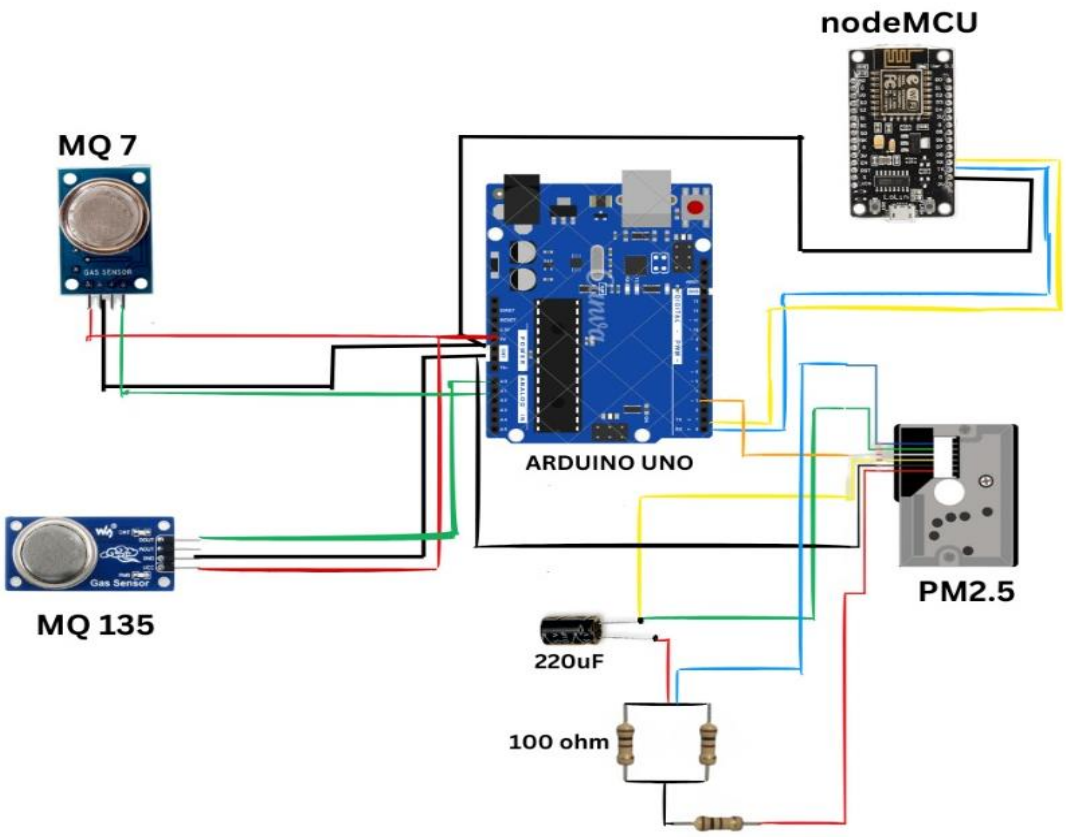

Fig 3.8 Components connection

3.3.3Models used for Prediction:

- Linear Regression Model:

  It is used for establishing relations between independent variables and dependent variables. When it comes to predicting air pollution, it is a very useful technique used

for figuring out how different environmental conditions affect the pollutant concentration. A linear regression model's general form can be written as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \varepsilon$$

Where:

Y is the dependent variable (e.g. pollutant concentration)

$X1, X2,...,Xn$ are the independent variables(e.g. Temperature, humidity etc)

$\beta 0, \beta 1, \beta 2,..., \beta n$ are the coefficients symbolizing the relationship between each independent variable and the dependent variable.

- Decision Tree Regressor:

In air pollution prediction using machine learning, decision tree regressors offer a simple yet powerful method. They construct a tree-like structure where each internal node represents a decision based on environmental factors, leading to leaf nodes that predict pollutant concentrations. In terms of construction, DT recursively split the feature space based on environmental variables. The splits are chosen to minimise variance in pollutant concentration within each subset.In prediction, the tree is traversed from root to leaf based on feature values (given X is an observation). At each internal node, a decision is made on which branch to follow. Ultimately the leaf node reached gives the predicted pollutant concentration. Mathematically it can be explained as:

$$Y = T(X)$$

Where:

Y represents the predicted pollutant concentration for a given observation X.

$T(X)$ represents the DT regressor function, which maps the feature vector X to the predicted pollutant concentration Y.

- Random Forest Regressor:

Random Forest consists of multiple decision trees trained on random subsets of features and data , reducing overfitting and improving generalisation. Each DT in the Random

forest is constructed independently, capturing different aspects of the data due to random selection of features at each split. Finally, predictions in Random Forest are formulated by aggregating the outcomes coming from individual tree predictions, basically using the average of all predictions for regression tasks AQI prediction. Mathematically it can be explained as:

$$Y = \frac{1}{N} \sum_{i=1}^{N} T_i(X)$$

Where:

Y represents the predicted pollutant concentration for a given observation *X*.

Ti(X) denotes the prediction from the *i*th decision tree in the Random Forest.

N is the total number of decision trees in the Random Forest

- Logistic Regression:

Logistic Regression is a widely used statistical method for binary classification tasks. It models the relationship between a binary dependent variable and one or more independent variables by estimating the probability of occurrence of a particular event. In logistic regression, the probability of the outcome variable y taking a value of 1 given the input features x1,x2,,,,,xn is modelled using the logistic function:

$$P(y = 1|x) = \frac{1}{1+e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n)}}$$

Where:

P is the probability of the positive class given input feature *x*

*β0, β1.....*are model coefficients.

- Decision Tree Classifier:

A Decision Tree Classifier recursively partitions the feature space into regions by selecting the best feature at each node to split the data. The splitting criterion aims to maximize the information gain or decrease in impurity, often measured by metrics like

Gini impurity or entropy. The decision tree can be represented as a series of if-else conditions, where each node represents a feature and each edge represents a decision rule based on that feature.

Mathematically:

$$h(x) = \sum_{i=1}^{m} (y_i \cdot 1(x \in R_i))$$

Where:

h(x) is predicted class for input

m is the number of terminal nodes.

Yi is the majority class in the region.

- Random Forest Classifier:

Random Forest Classifier is an ensemble learning method that aggregates predictions from multiple decision trees. Each tree is trained on a random subset of the training data and a random subset of the features.
The final prediction is obtained by averaging or taking the majority vote of the predictions from individual trees.

Mathematically:

$$y = \frac{1}{N} \sum_{i=1}^{N} h_i(x)$$

Where:

Y is the ensemble prediction for input *x*.

N is the number of tress in the forest

hi(*x*) is the prediction of the *i*th tree.

- K-Nearest Neighbors :

K-Nearest Neighbors (KNN) is a non-parametric, instance-based learning algorithm used for both classification and regression tasks. Given a query point *xq*, KNN identifies the K nearest data points in the training set based on a distance metric (e.g., Euclidean distance) and makes predictions based on the majority class (classification)

of those neighbors. Mathematically:

$$y_q = \text{argmax}_j \sum_{i=1}^{K} 1(y_i = j)$$

Where:

argmax*j* denotes the class with the highest count among the K nearest neighbors.

Yi is the class label of the *i*th nearest neighbor.

3.3.4 Implementation:

- In this section, different machine learning algorithms for regression and classifications are used to find the predicted AQI range based on the available dataset and sensor inputs from the hardware model.
- The dataset utilised for the purpose of training the Machine Learning Model is taken up from Kaggle [Air Quality Data India 2015 - 2020]. The dataset contains the timestamp values of concentration levels of various pollutants in different cities of India. The primary pollutants considered in the dataset are: PM2.5, PM10, NO2, NO, NOx, NH3, CO, O3, SO2 Toluene, Xylene and their respective concentration levels are represented in ug/m3. Dimension of the dataset is 737406 rows, 16 columns.
- To train the ML model using this dataset, libraries are imported. The libraries imported in the project are: NumPy, MatPlotLib, Pandas, Seaborn & Sklearn. The dataset is analysed thoroughly with the help of Pandas Dataframe and displayed in the console. To analyse the trends in the dataset pairplot, histogram & bar plot is utilised. The pairplot plots the pairwise relationships in the dataset; histogram is used to plot the frequency of values of each city in the dataset while bar plot is used to plot the concentration level of each pollutant against each city. The bar plots are further arranged in ascending order to identify the cities with highest concentration of pollutants. Conclusions are made based on the analysis graphs utilised. The dependent entity is the “Calculated AQI”. As this is a supervised machined learning technique we will be feeding both the inputs and the corresponding desired outputs of the given data for training of the models.
- Next step followed is Data Preprocessing step. Since the dataset contains null values, this step helps in detecting the number of null values in the dataset and identifying those

data points. Nearby values are interpolated to the nearest average value. Values that are not able to interpolate are dropped. Effective dimension of dataset: - 435513 rows × 16 columns.

- To optimise the dataset further, the concentration values of each pollutant is normalised to a definite level using mathematical functions and the resultant values are indexed [e.g. SO2 - > SOi]. For calculating AQI using the concentration values, the normalised values are rounded off and applied to a mathematical function which gives the result AQI. These AQI values can be utilised for regression models.
- Next step deals with integrating classification models with these AQI values. The AQI values are classified into different ranges and the resultant is indexed as AQI_bucket. This is followed by splitting the dataset into dependent and independent columns. The independent column is set as AQI_calculated while the dependent columns are normalised pollutant concentration values.
- After calculatingAQI values, they are classified as "Good, Satisfactory, Moderate, Poor, Very Poor, Severe" basedon some predefined ranges.

3.3.5 Proposed solution for integration of additional factors:

Incremental Learning Method:

- Incremental Learning is a methodology of machine learning where a ML Model learns and enhances its knowledge progressively, without forgetting previously acquired information. In traditional batch learning methods, the machine learning model is trained on the entirety of the dataset at once. However, in an incremental learning approach a ML model is trained batch wise as and when the data becomes available.
- This approach is particularly beneficial when the data trends remain changing over time or new parameters are being added to the initial dataset. This technique is also used when the used dataset is very large (~1TB in size) and has to be provided to the ML Model in a batch wise manner.
- The incremental learning model has the adaptive capability hence it can adapt to new parameters as well as taking into account the "knowledge" obtained from previously

trained data. This provides efficient use of resources and efficient learning from non-stationary data.

- Examples of Application of Incremental Learning Method: Recommendation Systems, Autonomous Vehicles, Sentiment Analysis
- Some of the Incremental Machine Learning Models are: Stochastic Gradient Descent Classifier (SGD Classifier), Online Support Vector Machines (SVM), Incremental Decision Trees such as Hoeffding Tree, Incremental Deep Learning Models etc. In our case, we have used SGDClassifier Model for incremental training purposes.
- Air Quality Prediction is a changing domain with respect to time. Industrialization, urbanization and traffic would not have played a major role in Air Quality some years back; but it has turned out to be a major leading factor for detecting quality of air. In some few years ahead, forest and green cover might also play a vital role in prediction of air quality. For such purposes, an incremental model comes handy for developing a proper ML based air quality prediction system over a traditional forecasting system.
- Air quality index depends on 3 major factors : Pollutant Concentration Level such as CO, N02, O3, PM2.5, PM10 etc.; weather conditions such as temperature, humidity, precipitation, wind speed, solar radiation etc. ; geographical (industrial, urban, rural, commercial area) and topographical (hilly, river plains, valley etc.) conditions of an area.

Methodology:

- To implement, SGD Classifier algorithm has been passed through two batches of training. For the first batch of training, the Kaggle dataset has been used consisting of data from 2015 - 2020. The dataset primarily consists of pollutant concentration levels of different cities of India such as : PM2.5, PM10, NO2, NOx, NH3, CO, SO2, O3, Benzene, Toluene, Xylene.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	City	Datetime	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene
2	Ahmedaba	########			1	40.01	36.37		1	122.07		0	0	0
3	Ahmedaba	########			0.02	27.75	19.73		0.02	85.9		0	0	0
4	Ahmedaba	########			0.08	19.32	11.08		0.08	52.83		0	0	0
5	Ahmedaba	########			0.3	16.45	9.2		0.3	39.53	153.58	0	0	0
6	Ahmedaba	########			0.12	14.9	7.85		0.12	32.63		0	0	0
7	Ahmedaba	########			0.33	15.95	10.82		0.33	29.87	64.25	0	0	0
8	Ahmedaba	########			0.45	15.94	12.47		0.45	27.41	191.96	0	0	0
9	Ahmedaba	########			1.03	16.66	16.48		1.03	20.92	177.21	0	0	0
10	Ahmedaba	########			1.47	16.25	18.02		1.47	16.45	122.08	0	0	0
11	Ahmedaba	########			2.05	13.78	16.08		2.05	15.14		0	0	0
12	Ahmedaba	########			2.27	13.87	16.73		2.27	14.12	99.17	0	0	0
13	Ahmedaba	########			1.73	12.87	14.63		1.73	13.26	91.67	0	0	0
14	Ahmedaba	########			1.72	14.15	15.55		1.72	17.2	95.92	0	0	0
15	Ahmedaba	########			1.85	15.74	17.62		1.85	18.78		0	0	0
16	Ahmedaba	########			0.95	15.94	16.18		0.95	19.16		0	0	0
17	Ahmedaba	########			0.87	17.28	18.32		0.87	17.83		0	0	0
18	Ahmedaba	########			0.8	19.04	20		0.8	16.14	187.62	0	0	0
19	Ahmedaba	########			0.47	21.24	22.7		0.47	11.93		0	0	0
20	Ahmedaba	########			0.53	25.63	27.42		0.53	14.99		0	0.33	0
21	Ahmedaba	########			0.47	16.22	16		0.47	13.66	187.42	0	0.23	0
22	Ahmedaba	########			0.83	16.5	17.52		0.83	13.28	96.08	0	0	0
23	Ahmedaba	########			0.85	17.97	18.18		0.85	12.23		0	0	0
24	Ahmedaba	########			1.08	15.52	15.4		1.08	10.5		0	0	0

Fig 3.9: Kaggle Dataset

- For the second batch of training, the dataset from Central Pollution Control Board, Bamunimaidan, Guwahati has been fetched. The dataset majorly consists of pollutant concentration levels from Railway Colony NFR Region (commercial / semi - industrial region) consisting of roughly 17,000 samples and also consists of weather conditions such as: 'temperature max', 'temperature min', 'temperature avg', 'feels like', 'dew', 'humidity', 'precipitation', 'precipitation type', 'snow', 'snow depth', 'wind gust', 'wind speed', 'wind direction' 'sea level pressure', 'cloud cover', 'visibility', 'solar radiation', 'solar energy', 'uv index', 'severe risk', 'sunrise', 'sunset', 'moonphase', thus increasing the samples to roughly around 30,000. Of all the factors, we have used "temperature", "dew", "uv index", "humidity", "wind speed", "precipitation", "solar radiation", "cloud cover" apart from all the pollutant concentration levels.

	A	B	C	D	E	F	G	H
1	CENTRAL POLLUTION CONTROL BOARD							
2	CONTINUOUS AMBIENT AIR QUALITY							
3							Date: Monday, Jun 10 2024	
4							Time: 04:23:43 PM	
5	Station	Railway Colony, Guwahati - PCBA						
6	Parameter	PM2.5,PM10,NO2,NO,NOx,NH3,SO2,CO,Ozone,Benzene,Toluene,Xylene						
7	AvgPeriod	1 Hours						
8	From	09-06-2022 T00:00:00Z 00:00						
9	To	10-06-2024 T16:11:59Z 00:00						
10								
11	Railway Colony, Guwahati - PCBA							
12			Railway Colony Guwah					
13	From Date	To Date	PM2.5	PM10	NO2	NO	NOx	NH3
14	09-06-2022 00:00	09-06-2022 01:00	39	81.36	2.47	5.51	5.37	6.77
15	09-06-2022 01:00	09-06-2022 02:00	40.5	81.86	2.43	5.33	5.17	6.75
16	09-06-2022 02:00	09-06-2022 03:00	41.47	81.07	2.71	5.45	5.34	8
17	09-06-2022 03:00	09-06-2022 04:00	64.61	108.86	2.45	5.4	5.24	8.9
18	09-06-2022 04:00	09-06-2022 05:00	26	45.07	2.39	5.44	5.24	9.64
19	09-06-2022 05:00	09-06-2022 06:00	19	40	2.62	5.45	5.33	8.91
20	09-06-2022 06:00	09-06-2022 07:00	20.94	38.57	2.57	5.36	5.26	7.12
21	09-06-2022 07:00	09-06-2022 08:00	41.47	53.93	2.37	5.54	5.35	6.48
22	09-06-2022 08:00	09-06-2022 09:00	21.79	30	2.36	5.46	5.21	6.39
23	09-06-2022 09:00	09-06-2022 10:00	5.79	10.86	2.46	5.49	5.31	7.04
24	09-06-2022 10:00	09-06-2022 11:00	4	9.07	2.49	5.38	5.19	7.56
25	09-06-2022 11:00	09-06-2022 12:00	11.57	15.5	2.54	5.43	5.29	7.37
26	09-06-2022 12:00	09-06-2022 13:00	15.43	20.64	2.48	5.42	5.22	6.78
27	09-06-2022 13:00	09-06-2022 14:00	None	None	2.41	5.59	5.36	6.44
28	09-06-2022 14:00	09-06-2022 15:00	None	None	2.49	5.45	5.29	5.21
29	09-06-2022 15:00	09-06-2022 16:00	None	None	2.56	5.36	5.31	5.08

Fig 3.10: CPCB Dataset

	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
1	tempmax	tempmin	temp	feelslikem	feelslikem	feelslike	dew	humidity	precip	preciprot	precipcov	preciptype	snow	snowdepti	windgust	windspeec	winddir	sealevelpr	cloudcove	visibility	solarradia	solarene
2	33	19.7	25.9	34.6	19.7	26.5	19.4	70.3	0.202	100	4.17	rain	0	0	35.3	22.6	31.5	1011.1	60.1	3.4	269.2	23
3	29	20.4	25.1	32	20.4	25.9	20.9	77.9	0.373	100	4.17	rain	0	0	34.2	16.3	33.6	1012.3	78.4	2.7	160	13
4	32	20.2	25.2	33.1	20.2	25.7	20.2	75.6	2.761	100	8.33	rain	0	0	29.5	22.3	244.8	1012.2	59.2	3.7	179.9	15
5	31	21.9	25.5	33.7	21.9	26.5	21.5	79.6	9.12	100	12.5	rain	0	0	28.4	19.4	318.4	1010.6	72.7	3.3	193.4	16
6	27	20.6	24	29.1	20.6	24.1	21.8	87.8	18.326	100	8.33	rain	0	0	39.2	35.2	277.7	1010.3	73.6	2.9	155	13
7	29	20.7	24.6	32	20.7	25.4	21.6	84.1	0.881	100	4.17	rain	0	0	43.9	13	301.1	1008.6	72	3.4	175.8	15
8	30.1	20.1	25.3	33.3	20.1	26.4	21	78.1	9.983	100	8.33	rain	0	0	45.7	16.6	265.1	1008.6	61.9	4	208	:
9	31	21	25.5	34.6	21	26.6	21.2	78.3	7.798	100	8.33	rain	0	0	38.5	18.4	16.9	1008	79.4	3.7	143.7	12
10	30	20.8	25	33.3	20.8	26	21.2	80.6	0.013	100	4.17	rain	0	0	37.1	20.5	40.4	1010	58.6	3.5	180.2	15
11	32	21.3	26.6	36.8	21.3	28.1	21	72.9	1.01	100	8.33	rain	0	0	29.2	16.6	43	1010	49.6	3.6	269.2	23
12	33	21.3	26.8	34.6	21.3	27.7	20.3	69.2	0	0	0		0	0	37.1	20.5	275.8	1009.6	58.9	3.4	254.8	:
13	27	22.1	24.3	28.8	22.1	24.6	21.3	83.4	1.067	100	8.33	rain	0	0	27	16.6	31.7	1008.9	84.4	3.3	236.5	20
14	27.2	21.7	23.7	29.1	21.7	23.8	21.7	89.1	1.42	100	8.33	rain	0	0	28.4	11.2	54.8	1007.9	75.8	2.5	196	16
15	27	21	23.8	30.1	21	24.1	22.6	93.3	15.856	100	8.33	rain	0	0	32	16.6	19.7	1007.5	84.6	2.9	158.6	13
16	30	21.7	25.5	34.2	21.7	26.7	22.7	85.5	11.394	100	12.5	rain	0	0	35.6	14.8	19.3	1006.5	79.2	3.2	140.6	12
17	32	22.1	26.6	39.2	22.1	28.6	23.2	82.5	5.798	100	4.17	rain	0	0	40.3	13	329.7	1007.4	81	3.4	144.8	12
18	33	23.1	27.4	40.6	23.1	30	23.2	79.1	1.399	100	8.33	rain	0	0	41.4	16.6	336.9	1008.1	66.9	3.2	184.4	15
19	32	22.2	27	37.9	22.2	29.5	23.2	80.7	0.7	100	8.33	rain	0	0	32.4	20.5	29	1008.6	55.1	3.3	252.7	21
20	33	22.5	25.9	39.3	22.5	27.3	21.9	79.4	0.56	100	4.17	rain	0	0	25.6	33.5	8.6	1008.2	61.3	3.3	243.8	20
21	32	21.7	25.7	34.2	21.7	26.3	19.8	71.8	0.282	100	12.5	rain	0	0	22.3	16.6	341.8	1009	68.1	3.3	235.4	20
22	31.6	20.3	26.4	36	20.3	28.1	21.6	75.9	1.077	100	4.17	rain	0	0	27.7	12.1	289.3	1010.4	66	3.4	259.3	22
23	33.6	22.2	27.5	37.5	22.2	29	21.7	73.1	0	0	0		0	0	24.8	18.4	356.8	1009.2	58.7	3.7	275.3	23
24	28	22.1	25	30.1	22.1	25.7	21.8	82.5	0.606	100	8.33	rain	0	0	29.5	14.8	16.3	1008.5	81.4	3.3	134.4	11
25	25	22.1	23.8	25	22.1	23.8	22.7	94	8.626	100	16.67	rain	0	0	34.9	16.6	31.4	1008.6	89.3	2.8	108.8	9
26	29	21.4	25.3	32.8	21.4	26.5	22.1	83.4	1.212	100	4.17	rain	0	0	24.1	14.2	31.7	1009.5	79.3	3.2	234.2	20
27	26	19.1	23.1	26	19.1	23.1	21.1	88.6	12.233	100	12.5	rain	0	0	33.8	16.1	60.9	1010.8	85.1	3	197.1	17
28	30	19.1	25.3	34.2	19.1	26.4	21.2	79.4	0	0	0		0	0	29.2	24.1	26.8	1011	65.3	3.8	241.1	20
29	27	22.6	24.4	30.1	22.6	24.6	22.4	89.1	0.793	100	12.5	rain	0	0	20.2	18.4	233.5	1009.5	85.5	3.1	157.4	13

Fig 3.11: Weather Dataset

- At first the pollutant concentration levels of different cities are converted to their respective sub-indices values, after proper data cleansing and exploratory data analysis (EDA). These values are fed to the SGDClassifier Model for the training purpose after segregating the dataset into training and testing dataset. The testing dataset is used for calculating the accuracy of the model. In the second batch of training, the data

preprocessing and data interpolation step is followed by adding of Guwahati based dataset consisting of pollutant concentration levels, weather factors etc.to the Classifier Model. This step is followed by calculation of accuracy of SGDClassifier Incremental Model. Thus, the SGDClassifier Model is applied in our Air Quality Prediction System.

SGDClassifier Model

```
[ ]  # Initialize the model
     model = SGDClassifier()

     # Initial training with dataset 1
     model.fit(X_train2, Y_train2)

     # Evaluate on dataset 1 test set
     accuracy1 = model.score(X_test2, Y_test2)
     print(f"Accuracy on dataset : {accuracy1}")
```

```
Accuracy on dataset : 0.5577593917710196
```

Fig 3.12: First batch training

```
[ ]  # Incremental learning with dataset 2
     model.fit(X_new_train, Y_new_train)

     # Evaluate on dataset 2 test set
     accuracy2 = model.score(X_new_test, Y_new_test)
     print(f"Accuracy on dataset 2: {accuracy2}")
```

```
Accuracy on dataset 2: 0.7317256740675949
```

Fig 3.13: Second Batch Training

3.4 Layout:

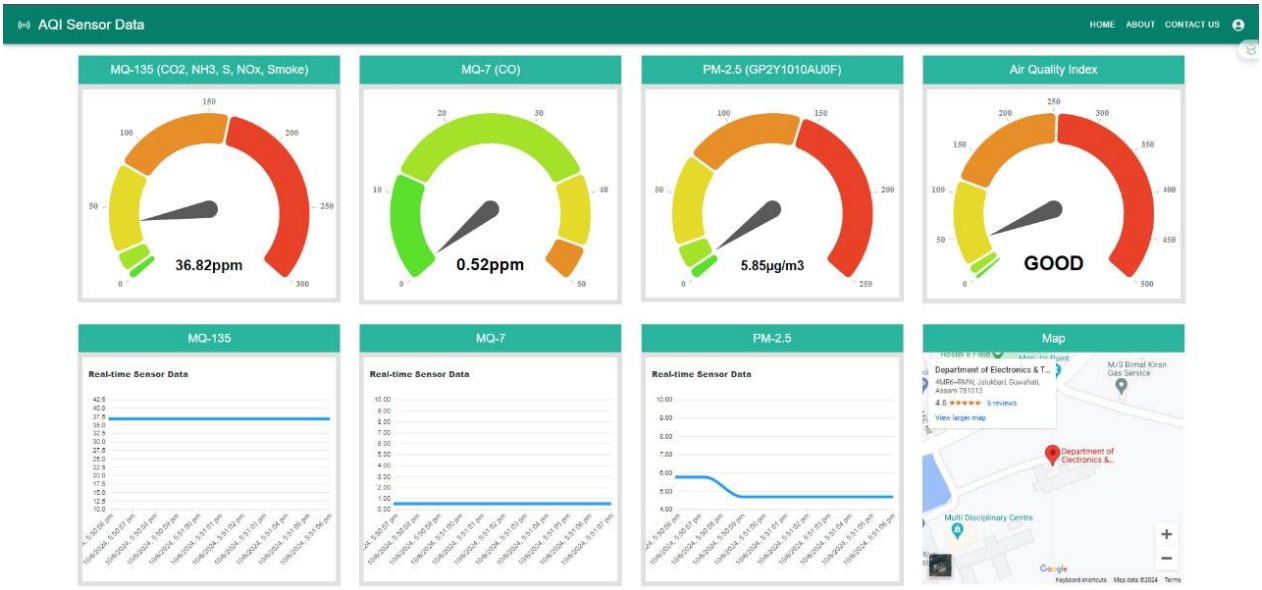

Fig 3.14 : Dashboard

The above figure shows the layout of dashboard. Here we have used gauges for individual sensors such as MQ-135; MQ-7; PM2.5 GP2Y1010AU0F. Below all the gauges we have also given graphs for individual sensors. We have also added AQI gauge which gives the current AQI value as per the data coming from the sensors. Map of the current location is also available.

CHAPTER 4

RESULT ANALYSIS

4.1 Introduction:

In this chapter, we will discuss the results we have obtained from various Machine Learning models that we have trained using prerecorded datasets. This dataset is then tested and the result is obtained. Also, we will discuss how we removed some of the errors that we found while testing and training our Machine Learning model.

For regression AQI_calculated is a function of SOi, NOi, NOx_i, NH3_i, COi, O3_i, Benzene, Toluene, Xylene AQI_calculated = f(SOi, NOi, NOx_i, NH3_i, COi, O3_i, Benzene, Toluene, Xylene) SOi, NOi, NOx_i, NH3_i, COi, O3_i, Benzene, Toluene, Xylene are independent columns and AQI_calculated is dependent column following the Training and testing dataset is prepared using test_size = 0.2. And random_state = 70 for regression algorithms.

For classification AQI_bucket_calculated is a function of SOi, NOi, NOx_i, NH3_i, COi, O3_i, Benzene, Toluene, Xylene AQI_bucket_calculated = f(SOi, NOi, NOx_i, NH3_i, COi, O3_i, Benzene, Toluene, Xylene) SOi, NOi, NOx_i, NH3_i, COi, O3_i, Benzene, Toluene, Xylene are independent columns AQI_bucket_calculated is dependent column Classification algorithms test_size = 0.33 and random_state = 70

4.2 Result Analysis:

(A)Hardware Result:

We integrated all the hardware components together and visualized the values of the individual sensors using graphs. To show sudden changes in the graph value we used gas lighter so that the concentration of pollutant changes . Resultant Graphs for the hardware sensors are:

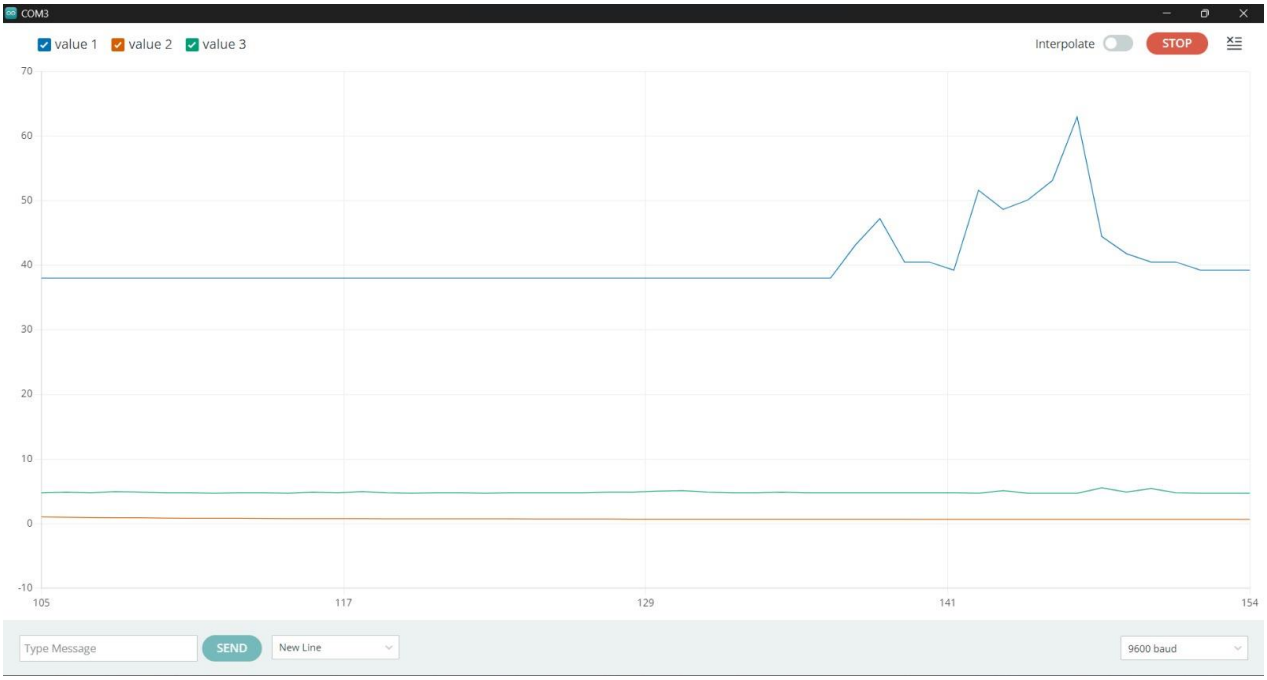

Fig 4.1 : Graph without sudden spike

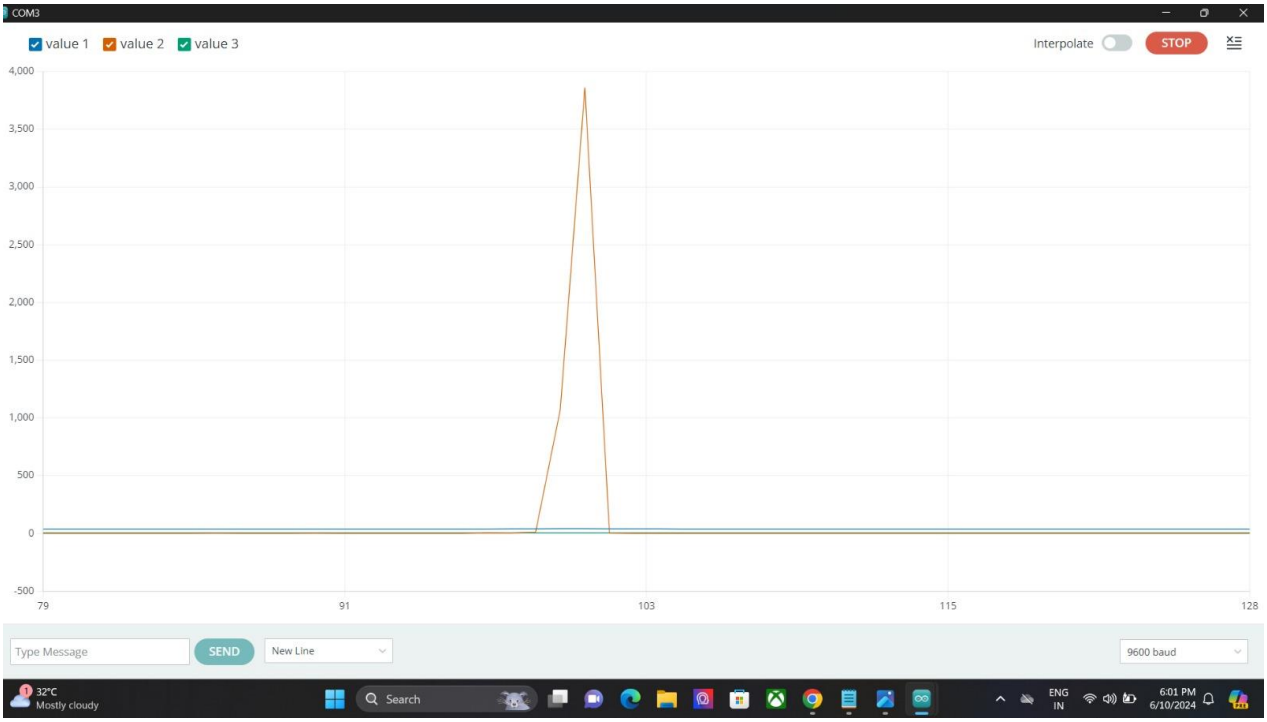

Fig 4.2 : Graph with spike

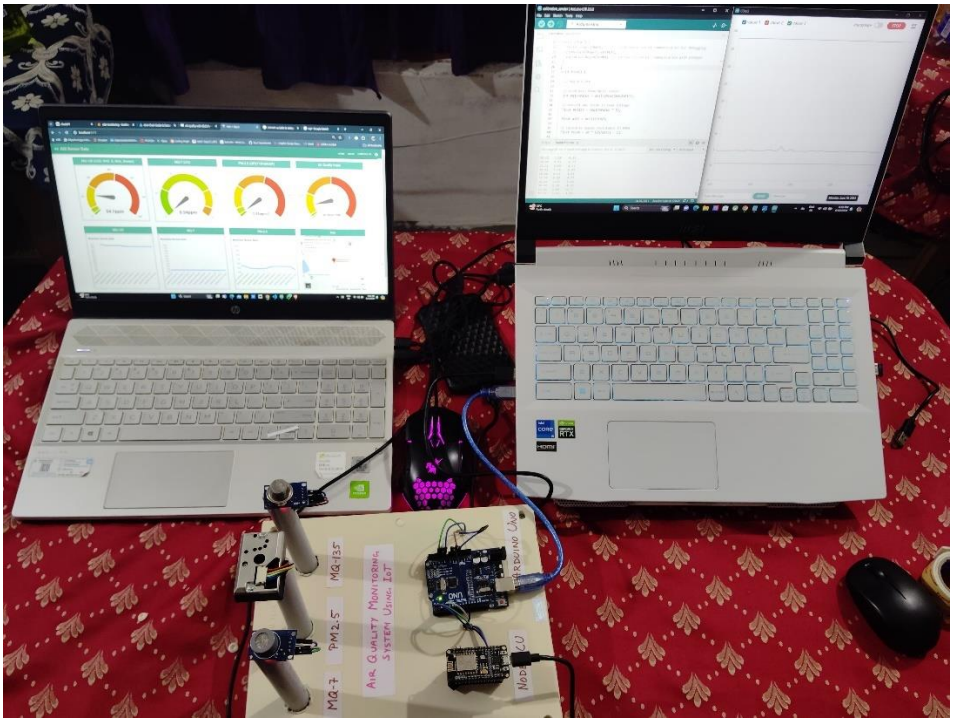

Fig 4.3 Hardware Model top view

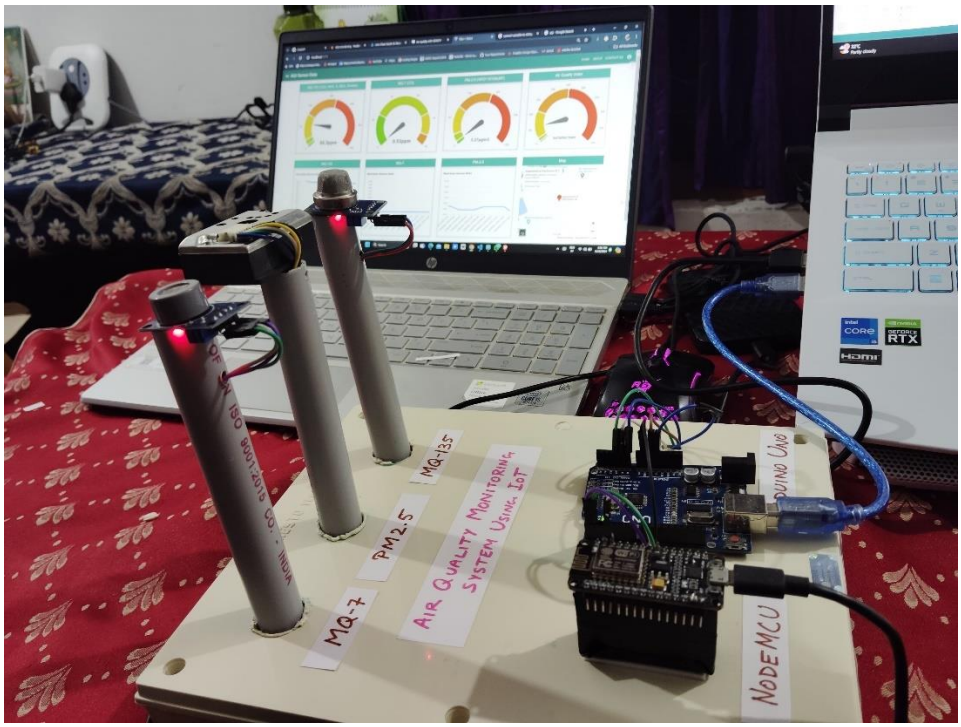

Fig 4.4 Hardware Model

(B) Software Result:

The following graph shows the presence of the different components of the pollutant present in the data set corresponding to different cities.

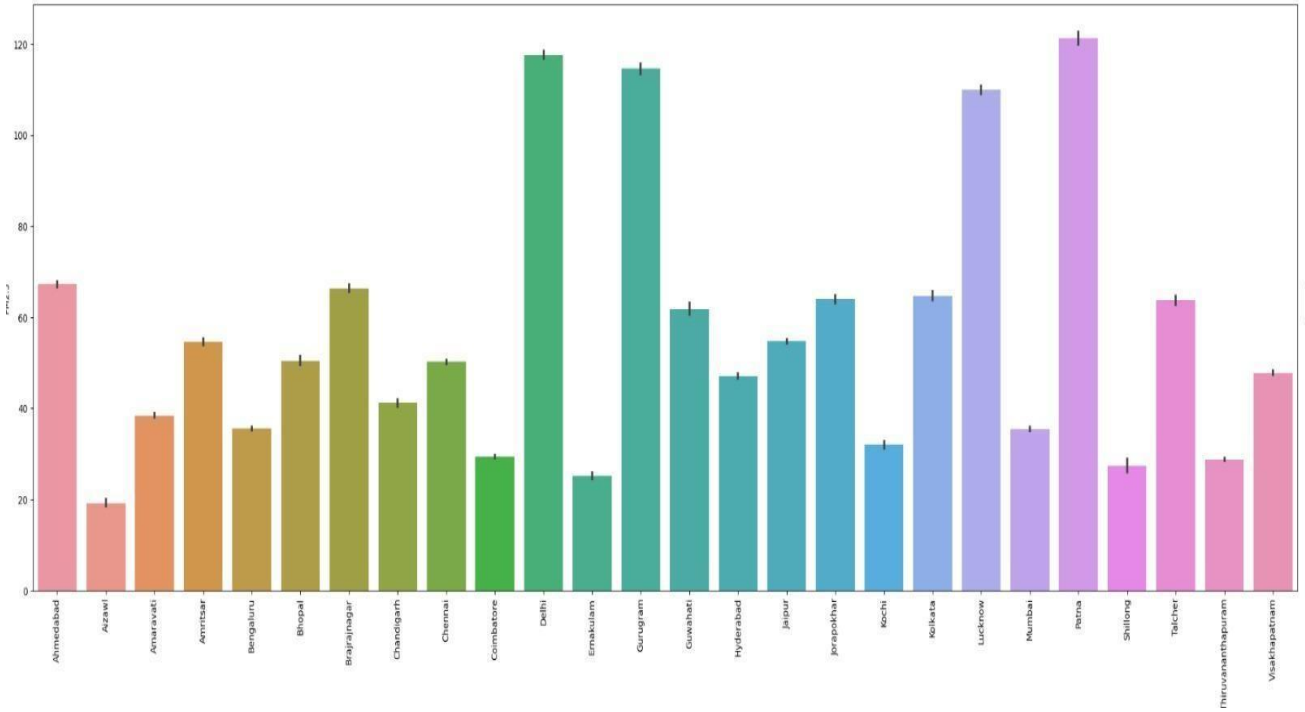

Fig 4.5: Level of PM2.5 in different cities

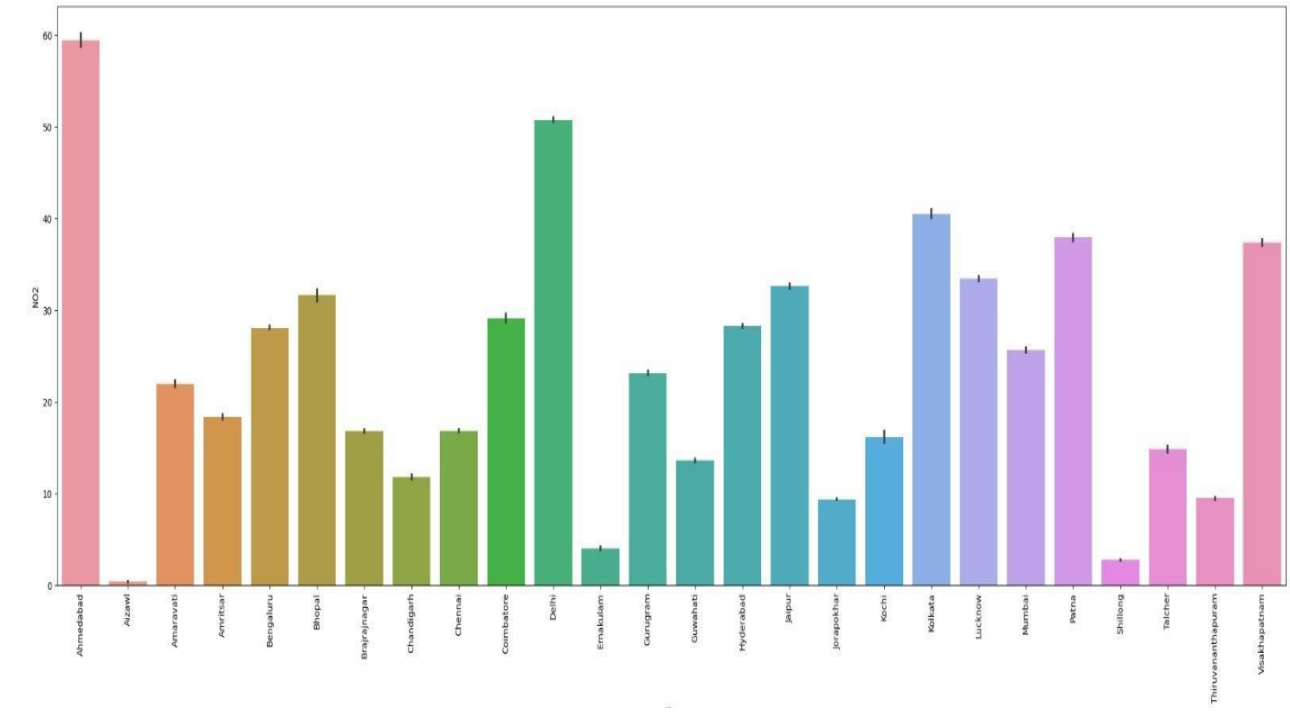

Fig 4.6: Level of NO2 in different cities

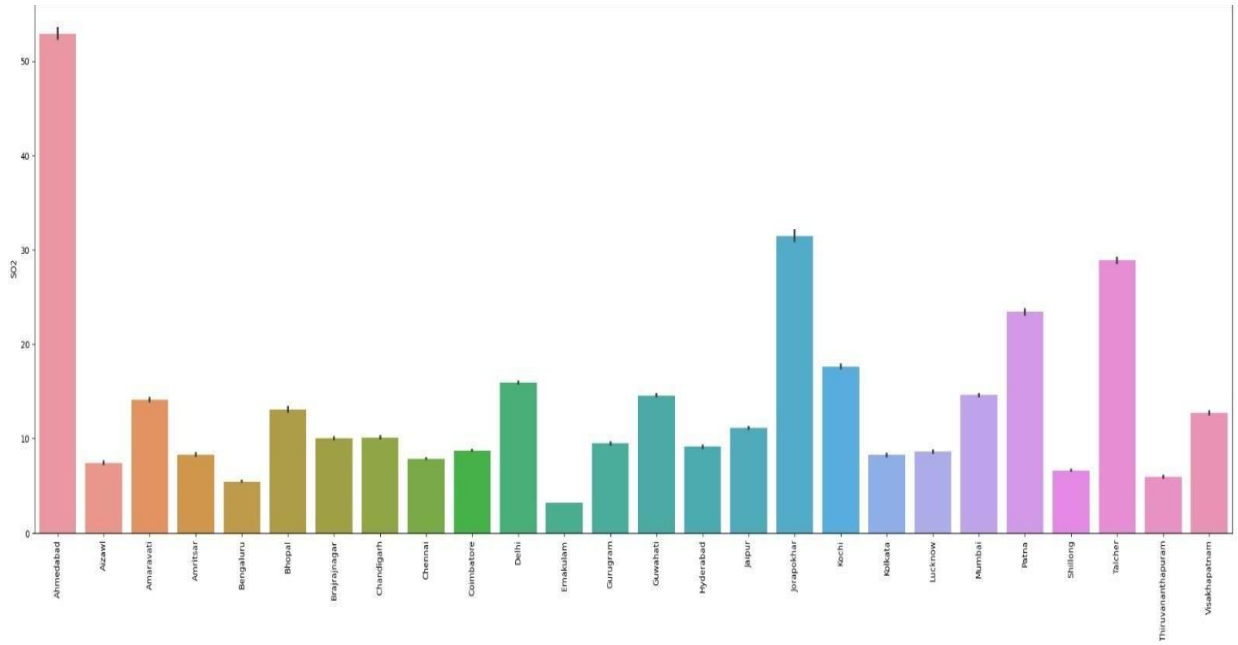

Fig 4.7: Level of SO2 in different cities

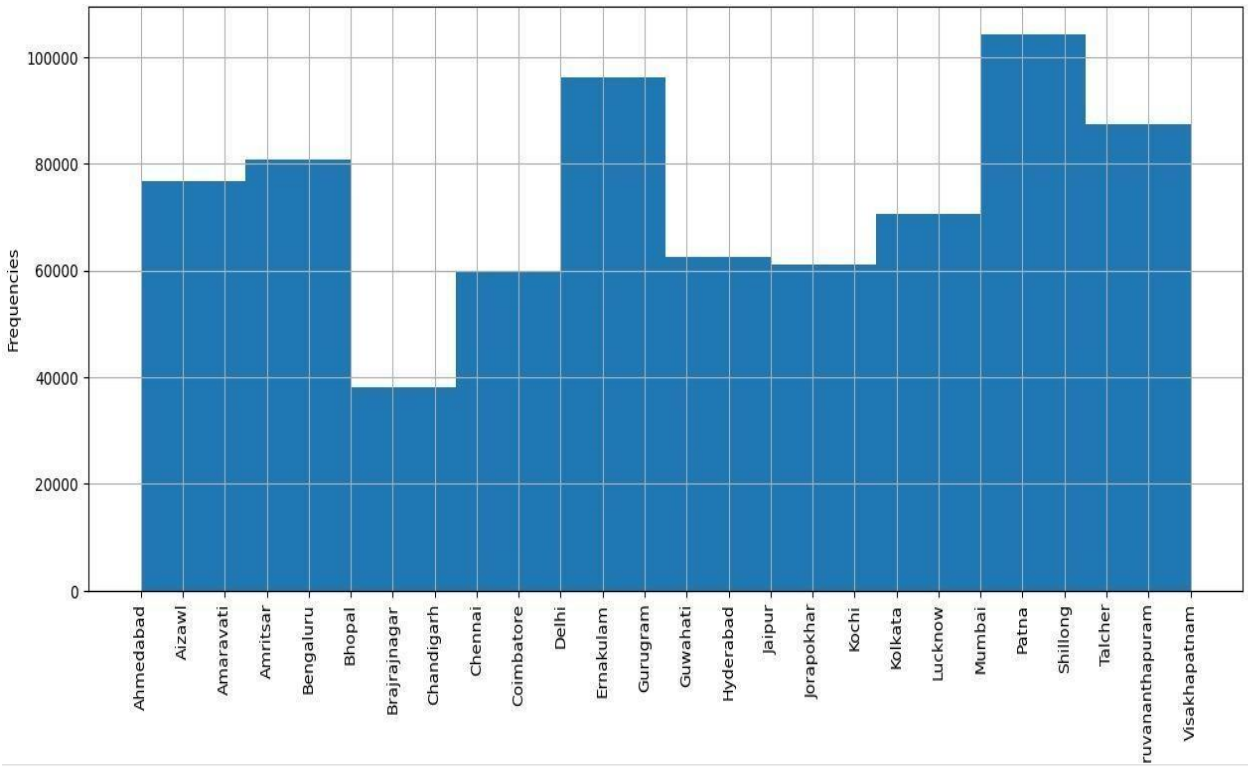

Fig 4.8: Frequency of data of different cities

Different error-checking parameters such as Root Mean Square Error (RMSE), R-squared, and Mean absolute errors (MAE) are used to check the performance of our regression models. Similarly, the parameters like Accuracy and Kappa score are used to check the performance of our classification models. Some brief descriptions of these performance parameters are:

- **Root mean square error (RMSE)** or root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions. It shows how far predictions fall from measured true values using Euclidean distance.

- **R-squared (R^2)** is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable in a regression model.

- **Mean absolute Error (MAE)** takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group. MAE can also be referred to as L1 loss function.

- **Accuracy** is the percentage of correct classifications that a trained machine learning model achieves, i.e., the number of correct predictions divided by the total number of predictions across all classes.

- **Cohen's Kappa score** can be defined as the metric used to measure the performance of machine learning classification models based on assessing the perfect agreement and agreement by chance between the two raters (a real-world observer and the classification model).

Following are the results that are derived from the machine learning models that are trained on the datasets that are collected from different resources. The performance of the model is tested on both the train and the test data set.

Table 4.1: Performance of various Regression Models

Model	**RMSE**	**RSquared**	**MAE**
Linear Regression	20.351 (Train) 19.830 (Test)	0.9638 (Train) 0.9661 (Test)	12.88774
Decision Tree Regressor	0.0 (Train) 2.6837 (Test)	1.0 (Train) 0.9993 (Test)	0.30170
Random Forest Regressor	1.055 (Train) 2.184 (Test)	0.9999 (Train) 0.9995 (Test)	0.24538

Table 4.2: Performance of various Classification Models

Model	**Accuracy**	**Kappa Score**
Logistic regression	0.5381 (Train) 0.5362 (Test)	0.3533
Decision tree classifier	1.0 (Train) 0.9994 (Test)	0.9992
Random Forest Classifier	1.0 (Train) 0.9981 (Test)	0.9974
K- Nearest Neighbours	0.9672 (Train) 0.9418 (Test)	0.9213

Table 4.3 Performance of Stochastic Gradient Descent Classifier

Stochastic Gradient Descent Classifier	Accuracy after 1st Batch Training	Accuracy after 2nd Batch Training(Final)
	0.5577	0.7317

4.3 Significance of result obtained:

The significance of the result obtained above from the regression and classification models lies in their ability to accurately predict outcomes based on the input data. The result obtained can be compared with the results of existing research work done by Hemanth Karnati[11](Pg.13-15)

Following is the comparison between the results:

Table 4.4: Performance comparison between our and referenced Regression Model

Model	**Our Model**			**Hemanth Karnati's Model**			**Inference**
	RMSE	**RSquared**	**MAE**	**RMSE**	**RSquared**	**MAE**	
Linear Regression	20.351 (Train) 19.830 (Test)	0.9638 (Train) 0.9661 (Test)	12.88774	26.95	0.91	18.98	Better Results as compared to existing RMSE RSquared And MAE values
Decision Tree Regressor	0.0 (Train) 2.6837 (Test)	1.0 (Train) 0.9993 (Test)	0.30170	30.21	0.89	20.48	Better Results as compared to existing RMSE RSquared And MAE values
Random Forest Regressor	1.055 (Train) 2.184 (Test)	0.9999 (Train) 0.9995 (Test)	0.24538	22.00	0.94	15.02	Better Results as compared to existing RMSE RSquared And MAE values

Table 4.5: Performance comparison between our and referenced Classification Model

Model	**Our Model**		**Hemanth Karnati's Model**		**Inference**
	Accuracy	**Kappa Score**	**Accuracy**	**Kappa Score**	
Logistic regression	0.5381 (Train) 0.5362 (Test)	0.3533	0.40	0.41	Better Accuracy and Kappa Score
Decision tree classifier	1.0 (Train) 0.9994 (Test)	0.9992	0.71	0.71	Better Accuracy and Kappa Score
Random Forest Classifier	1.0 (Train) 0.9981 (Test)	0.9974	0.80	0.80	Better Accuracy and Kappa Score
K- Nearest Neighbours	0.9672 (Train) 0.9418 (Test)	0.9213	0.80	0.80	Better Accuracy and Kappa Score

4.4 Conclusion:

In conclusion, the analysis of the regression and classification models reveals valuable insights into the performance. The linear regression model demonstrates good predictive capabilities on both training and testing datasets, indicating a solid understanding of the underlying patterns in the data. Also, the results ofthe trained model are compared with the results of the existing research

work model and found that the regression and classification models have better results in terms of RMSE, R Squared, MAE, Accuracy, and Kappa Score which shows that we were able to improve the overall Machine Learning Algorithm which will further give better prediction results. Also, The incremental learning approach might prove to be a viable method for predicting air quality index and pollution concentration levels if provided with suitable dataset as well as implementing sophisticated ML models. Thus the SGD Classifier Model has been implemented as a proposed solution to add additional factors in a sequential batch wise learning approach for providing more accurate and efficient results.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE OF WORK

5.1Work Summary:

5.1.1 Problem Statement/ Objective:

The project aims to develop an Air Quality Monitoring and Prediction model using the Internet of Things (IoT) and Machine Learning (ML) to take advantage of advanced technologies so that we can address air pollution challenges and improve the overall quality of air in our environment.
The key objectives of the project include:

- Real-time Monitoring: Implement an IoT-based system for real-time monitoring of air quality parameters, including particulate matter (PM), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO) etc
- Data Quality Assurance: Develop mechanisms to ensure the accuracy, reliability, and consistency of air quality data collected from distributed sensors.
- Predictive Modeling: Utilize machine learning algorithms to analyze historical and real- time data to develop accurate predictive models for air quality levels, considering the influence of dynamic environmental factors.

The successful implementation of an IoT and ML-based air quality monitoring and prediction system will provide a comprehensive, real-time understanding of air quality, allowing proactive measures to address pollution issues and, ultimately, improve public health and environmental sustainability.

5.1.2 Brief Summary of the work:

The use of IoT (Internet of Things) and ML (Machine Learning) for air pollution monitoring and prediction has emerged as a cutting-edge approach to addressing the challenges associated with deteriorating air quality. IoT enables real-time data collection on various air pollutants such as particulate matter, ozone, and nitrogen dioxide through the deployment of sensor networks and connected devices. This integration of IoTand ML not only improves the precision of pollution monitoring but also contributes to the development of new technologies.

5.2 Conclusion:

In conclusion, the project on "Air Pollution Monitoring and Prediction using IoT and Machine Learning" represents a significant stride towards addressing the complex challenges posed by air pollution in urban environments. By integrating state-of-the-art technologies such as the Internet of Things (IoT) and advanced Machine Learning (ML) algorithms, the project has laid the foundation for a comprehensive and dynamic system capable of providing real-time air quality insights and predictive analytics. The Machine Learning models developed for air quality prediction showcase the project's commitment to leveraging data-driven approaches for environmental monitoring. By exploring regression models, ensemble methods etc, the project has aimed at capturing the intricate relationships within the collected data. The integration of historical and real-time data into these models provides a basis for informed predictions and proactive decision-making.

5.3 Future Scope:

5.3.1 Enhanced Sensor Integration and Network Expansion:

A significant area for future development is the integration of advanced sensors capable of detecting a broader range of air pollutants and environmental parameters. By incorporating sensors that measure additional pollutants such as nitrogen oxides (NOx), sulfur oxides (SOx), and ozone, the system can provide a more comprehensive analysis of air quality. Expanding the sensor network to cover larger geographical areas, including rural and remote locations, will yield more granular data. This expansion will enable precise monitoring and improve the system's overall effectiveness in diverse environmental contexts.

5.3.2 *Advanced Machine Learning Models and Real-time Analytics:*

Implementing more sophisticated machine learning models, such as deep learning and ensemble methods, is another promising direction for future work. These models can enhance the accuracy and robustness of air quality predictions. Developing real-time analytics capabilities will enable immediate alerts and insights for users, fostering proactive responses to pollution events. Integrating adaptive algorithms that continuously learn from new data will ensure the system

remains accurate and relevant over time. Additionally, exploring edge computing can reduce latency and enhance data processing efficiency, making the system more responsive and reliable.

5.3.3 *User Interface and Community Engagement:*

Improving the user interface and developing mobile applications will enhance user engagement and accessibility. A more intuitive and user-friendly interface will enable users to interact with the system effectively, understanding air quality data and predictions easily. Community engagement initiatives, such as involving local communities in data collection and validation, will improve data accuracy and foster a sense of ownership and responsibility towards air quality management. Public awareness campaigns and educational tools integrated into the platform can help users make informed decisions about their health and environment.

5.3.4 Integration with Smart City Initiatives:

Integrating the air pollution monitoring and prediction system with smart city initiatives can amplify its impact. By collaborating with municipal authorities, the system can provide real-time data to optimize traffic management, control industrial emissions, and enhance urban planning. This integration can lead to more efficient pollution control measures and contribute to the development of healthier and more sustainable urban environments. Such integration will also support the creation of data-driven policies for better urban management and environmental protection.

5.3.5 Predictive Maintenance and Automated Responses:

Incorporating predictive maintenance algorithms will ensure the long-term reliability and accuracy of the deployed sensors. These algorithms can predict when a sensor is likely to fail or require recalibration, allowing for timely maintenance and reducing downtime. Additionally, developing automated response systems that trigger specific actions, such as activating air purifiers or issuing health advisories during high pollution levels, can significantly enhance the system's effectiveness in protecting public health. These automated responses can help mitigate the adverse effects of pollution more efficiently.

5.3.6 Data Privacy and Security Enhancements:

As the system collects and processes large amounts of data, ensuring data privacy and security becomes paramount. Future work can focus on implementing robust encryption methods and secure data transmission protocols to protect sensitive information. Developing policies and frameworks for data governance, including user consent management and compliance with relevant regulations, will enhance trust and acceptance of the system among users and stakeholders. Ensuring data integrity and confidentiality will be crucial in maintaining the credibility and reliability of the system.

REFERENCES

[1] “Air Quality Prediction and Monitoring using Machine Learning Algorithm based IoT Sensor-A Researcher's Perspective”. ICCES 2021, DOI: 10.1109/ICCES51350.2021.9489153

[2] “Air Pollution Monitoring and Prediction Using IOT and Machine Learning”. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 12 (2), 2021, 60-65

[3] “Air pollution monitoring and prediction using IoT”. DOI: 10.1109/ICICCT.2018.8473272

[4] “Air Quality Monitoring And Prediction Using IOT And Machine Learning Approaches”. DOI: 10.29322/ijsrp.12.03.2022.p12307

[5] “Air Quality Monitoring and Forecasting System Using IoT and Machine Learning Techniques”. DOI: 10.1109/GTSD54989.2022.9988756

[6] “Real Time Air Pollution Prediction in Urban Cities using Deep Learning Algorithms and IoT”. DOI: 10.1109/ICCES54183.2022.9835991

[7] “A Bagging-GBDT ensemble learning model for city air pollutant concentration prediction”. DOI: 10.1088/1755-1315/237/2/022027

[8] “Detection and Prediction of Air Pollution using Machine Learning Models”. DOI: 10.14445/22315381/IJETT-V59P238

[9] “IoT enabled Environmental Air Pollution Monitoring and Rerouting system using Machine learning algorithms”. DOI: 10.1088/1757-899X/955/1/012005

[10] “Machine Learning Techniques for Air Quality Forecasting and Study on Real-Time Air Quality Monitoring”. DOI: 10.1109/ICCUBEA.2017.8463746

[11] “IoT-Based Air Quality Monitoring System with Machine Learning for Accurate and Real-time Data Analysis”. DOI: 10.48550/arXiv.2307.00580

ANNEXURE

Arduino code

```
#include <SoftwareSerial.h>

const int mq135Pin = A0;  // Analog pin for MQ135 sensor
const int mq7Pin = A1; // Analog pin for MQ7 sensor
const int pm25Pin = A2; // Analog pin for PM2.5 GP2Y1010 sensor
const int ledPower = 2; // Connect led driver pins of dust sensor to arduino D2
const int esp8266Tx = 2;  // Tx pin on Arduino Uno
const int esp8266Rx = 3;  // Rx pin on Arduino Uno

SoftwareSerial espSerial(esp8266Rx, esp8266Tx); // Create a SoftwareSerial object

int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;             // For PM2.5 Sensor

float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;

void setup() {
  Serial.begin(9600);     // Initialize serial communication for debugging
  pinMode(ledPower, OUTPUT);
  espSerial.begin(9600);  // Initialize serial communication with ESP8266
}

void loop() {

  // MQ135 BLOCK

  // Read data from MQ135 sensor
  int mq135Value = analogRead(mq135Pin);

  // Convert ADC Value to Load Voltage
  float Vrl135 = (mq135Value * 5);

  float a135 = Vrl135/1023;

  // Calculate Sensor Resistance in kOhm
  float Rs135 = 20 * ((5/a135) - 1);

  // Sensor Resistance for ambient environment
  float Ro135 = 127.19;        // Ro = 217.91
```

```
// R_coeff
float R_coeff_135 = (Rs135/Ro135);

// ppm eqn: PPM = a * (Rs/Ro)^b
float exponent_135 = pow(R_coeff_135,-2.78054);     // a = 121.4517, b = -2.78054
float ppm_MQ135 = exponent_135 * 121.4517;

// MQ7 BLOCK

// Read data from MQ7 sensor
int mq7Value = analogRead(mq7Pin);

// Convert ADC Value to Load Voltage
float Vrl7 = (mq7Value * 5);
float a7 = Vrl7/1023;

// Calculate Sensor Resistance in kOhm
float Rs7 = 10 * ((5/a7) - 1);

// Sensor Resistance for ambient environment
float Ro7 = 68.69;

// R_coeff
float R_coeff_7 = (Rs7/Ro7);

// ppm eqn: PPM = (Rs/(a*Ro))^(1/b)
float ppm_7 = pow(R_coeff_7/1.0 , -2.22);     // a = 1.0, b = -0.45

// PM2.5 BLOCK

digitalWrite(ledPower, LOW); // power on the LED
delayMicroseconds(samplingTime);

int pm25Value = analogRead(pm25Pin); //read the PM2.5 Sensor value

delayMicroseconds(deltaTime);
digitalWrite(ledPower, HIGH); // turn the LED off
delayMicroseconds(sleepTime);

calcVoltage = pm25Value * 5.0; //0-5V  mapped to 0 - 1023 values
float a25=calcVoltage/1024.0;
dustDensity = 17*(a25 - 0.1); //dust density in ug/m3
```

```
  // Print the calibrated values
  //Serial.print("MQ135: ");
  Serial.print(ppm_MQ135);
  Serial.print("\t");
  delay(1000);

  //Serial.print("MQ7: ");
  Serial.print(ppm_7);
  Serial.print("\t");
  delay(1000);

  //Serial.print("PM2.5: ");
  Serial.println(dustDensity);

  // Send data to ESP8266
  espSerial.println(ppm_MQ135);
  espSerial.println(ppm_7);
  espSerial.println(dustDensity);

  // Wait for a while before sending the next data
  delay(2000);
}
```

NodeMCU Code

```
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

const int esp8266Tx = 2;  // Tx pin on ESP8266 (connected to Rx pin on Arduino Uno)
const int esp8266Rx = 3;  // Rx pin on ESP8266 (connected to Tx pin on Arduino Uno)

SoftwareSerial espSerial(esp8266Rx, esp8266Tx); // Create a SoftwareSerial object

String apiKey = "AIzaSyBq1zWsRNgVcK37Lu8hovcT6X4wvRjHjpU";  // Enter your API key here
const char *ssid = "Redmi Note 7S";  // Enter your Wi-Fi Name
const char *pass = "b914fede09ba";  // Enter your Wi-Fi Password
const char *databaseURL = "https://aqi-monitoring-b45d4-default-rtdb.asia-
southeast1.firebasedatabase.app/";
WiFiClient client;

// Define Firebase Data object
FirebaseData fbdo;
```

```
FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;
bool signupOK = false;

void setup() {
  Serial.begin(9600);     // Initialize serial communication for debugging
  espSerial.begin(9600);  // Initialize serial communication with Arduino Uno

  Serial.println("Connecting to Wi-Fi");

  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  // Assign the API key
  config.api_key = apiKey;

  // Assign the RTDB URL
  config.database_url = databaseURL;

  // If using email and password for authentication, uncomment the following lines and provide valid
credentials
  auth.user.email = "paulabhiraj56@gmail.com";
  auth.user.password = "abhiraj11";

  // Sign up (for anonymous sign-in, leave email and password as empty strings)
  if (Firebase.signUp(&config, &auth, "", "")) {
    Serial.println("Sign up successful");
    signupOK = true;
  } else {
    Serial.printf("Sign up failed: %s\n", config.signer.signupError.message.c_str());
  }

  // Assign the callback function for the long running token generation task
  config.token_status_callback = tokenStatusCallback; // see addons/TokenHelper.h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
```

```
}

void loop() {
  if (espSerial.available()) {
    Serial.println("Data available from ESP8266");

    // Read data from Arduino Uno
    String data1 = espSerial.readStringUntil('\t');
    Serial.print("MQ135: ");
    Serial.print(data1);
    Serial.print("\n");

    String data2 = espSerial.readStringUntil('\t');
    Serial.print("MQ7: ");
    Serial.print(data2);
    Serial.print("\n");

    String data3 = espSerial.readStringUntil('\t');
    Serial.print("PM2.5: ");
    Serial.println(data3);

    // Convert data to integers
    float mq135Value = data1.toFloat();
    float mq7Value = data2.toFloat();
    float pm25Value = data3.toFloat();

    // Print converted data
    Serial.print("Converted MQ135: ");
    Serial.println(mq135Value);
    Serial.print("Converted MQ7: ");
    Serial.println(mq7Value);
    Serial.print("Converted PM2.5: ");
    Serial.println(pm25Value);

    if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 1000 || sendDataPrevMillis == 0)) {
      sendDataPrevMillis = millis();

      // MQ135 Sensor
      if (Firebase.RTDB.setInt(&fbdo, "Sensor/MQ135", mq135Value)) {
        Serial.print("MQ135 value sent: ");
        Serial.println(mq135Value);
      } else {
        Serial.println("Failed to update MQ-135");
        Serial.println("REASON: " + fbdo.errorReason());
      }

      // MQ7 Sensor
```

```
    if (Firebase.RTDB.setInt(&fbdo, "Sensor/MQ7", mq7Value)) {
      Serial.print("MQ7 value sent: ");
      Serial.println(mq7Value);
    } else {
      Serial.println("Failed to update MQ7");
      Serial.println("REASON: " + fbdo.errorReason());
    }

    // PM2.5 Sensor
    if (Firebase.RTDB.setInt(&fbdo, "Sensor/PM25", pm25Value)) {
      Serial.print("PM2.5 value sent: ");
      Serial.println(pm25Value);
    } else {
      Serial.println("Failed to update PM2.5");
      Serial.println("REASON: " + fbdo.errorReason());
    }
   }

   client.stop();
   delay(500);
  }
}
```

Dashboard Link

https://aqi-dashboard-aec.netlify.app/