IOT AND MACHINE LEARNING-BASED SYSTEM FOR PREDICTING AND MONITORING INDOOR AIR QUALITY

A PROJECT REPORT

Submitted by

AAKASH D	**511321106001**
CHANDRASEKRAN S	**511321106005**
DINESH A	**511321106010**
YUKENDIRAN J	**511321106306**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

KINGSTON ENGINEERING COLLEGE

ANNA UNIVERSITY:CHENNAI 600 025

MAY 2025

ANNA UNIVERSITY:CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"IOT AND MACHINE LEARNING-BASED SYSTEM FOR PREDICTING AND MONITORING INDOOR AIR QUALITY"** is the Bonafide work of **AAKASH D (511321106001), CHANDRASEKRAN S (511321106005), DINESH A (511321106010), YUKENDIRAN J (511321106306)** who carried out the project work under my supervision during the academic year 2024 – 2025.

SIGNATURE	**SIGNATURE**
Mrs. M. RATHIKA. M.E., Ph.D.,	**Dr. A.VENKATESAN. M.E., Ph.D.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Electronics and Communication Engineering,	Electronics and Communication Engineering,
Kingston Engineering College,	Kingston Engineering College,
Vellore - 632059.	Vellore - 632059.

CERTIFICATE OF EVALUATION

COLLEGE NAME	KINGSTON ENGINEERING COLLEGE
BRANCH	ELECTRONICS AND COMMUNICATION ENGINEERING
SEMESTER/BATCH	VIII/2021-2025
PROJECT TITLE	IOT AND MACHINE LEARNING-BASED SYSTEM FOR PREDICTING AND MONITORING INDOOR AIR QUALITY
SUBJECT CODE	EC3811

S. NO.	NAME	PROJECT TITLE	SUPERVISOR NAME
1	AAKASH D (511321106001)	IOT AND MACHINE LEARNING-BASED SYSTEM FOR PREDICTING AND MONITORING INDOOR AIR QUALITY	Dr. A.VENKATESAN, M.E., Ph.D.,
2	CHANDRASEKRAN S (511321106005)		
3	DINESH A (511321106010)		
4	YUKENDIRAN J (511321106306)		

The report of the project work submitted by the above students in partial fulfillment for the award of **BACHELOR OF ENGINEERING DEGREE IN ELECTRONICS AND COMMUNICATION OF ANNA UNIVERSITY** The project report was submitted for viva voce held on ______________ at Kingston Engineering College.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

At this pleasing moment of having successfully completed our project, we wish to convey our sincere thanks and gratitude to the management of the institution and our specific heartfelt thanks and gratitude to our beloved chairman, **Thiru. D. M. KATHIRANAND, M.B.A., (USA),** who provided all the facilities to us.

We would like to express our sincere thanks to **Dr. U.V.ARIVAZHAGU,M.E., Ph.D.,** Principal, Kingston Engineering College, for allowing us to do our project and offering adequate support in completing our project.

We are grateful to our Head of the Department and our project coordinator, **Mrs.M.RATHIKA,M.E.,Ph.D.,** Kingston Engineering College, for her constructive suggestions, support and encouragement during our project.

We extend our earnest and sincere thanks and gratitude to **Dr. A.VENKATESAN, M.E.,Ph.D.,** our project supervisor, Assistant Professor, Kingston Engineering College, for her kind support, direction and guidance in completing the project successfully.

We would like to express our sincere thanks to all our department teaching and non - teaching faculty members, family members and friends for their timely support during the process of our project.

ABSTRACT

Air pollution poses serious health risks, particularly in indoor environments where people spend most of their time. Pollutants like carbon monoxide, nitrogen dioxide, and particulate matter contribute to respiratory and cardiovascular diseases. Traditional monitoring systems are often expensive, limited in scope, and lack real-time insights.

This project proposes a cost-effective, IoT and Machine Learning-based system to monitor and predict indoor air quality. The system measures key parameters such as PM2.5, CO, NO_2, SO_2, NH_3, temperature, and humidity using sensors like MQ-135, MQ-2, DHT11, and PM2.5 integrated with Arduino UNO and NodeMCU microcontrollers.

To enhance predictive capability, the collected data is analyzed using Random Forest and Support Vector Machine (SVM) algorithms. The SVM model achieved up to 99% accuracy, while Random Forest delivered reliable predictions across multiple pollutant levels.This integrated approach enables real-time monitoring and accurate forecasting of air quality trends, supporting timely health and safety interventions in indoor spaces.

KEYWORDS:

IoT, Indoor Air Quality, Machine Learning, SVM, Random Forest, AQI Prediction, Arduino UNO, NodeMCU, Gas Sensors, Real-Time Monitoring

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Air pollution poses a significant threat to both environmental sustainability and public health, contributing to approximately 1.3 million deaths globally each year, according to the World Health Organization (WHO). It not only degrades air quality but also leads to serious consequences such as global warming, acid rain, respiratory diseases, and even exacerbation of conditions like COVID-19. In rapidly developing countries like India, population growth and urbanization have further intensified pollution-related challenges. To address these concerns, real-time air quality monitoring and forecasting have become essential.

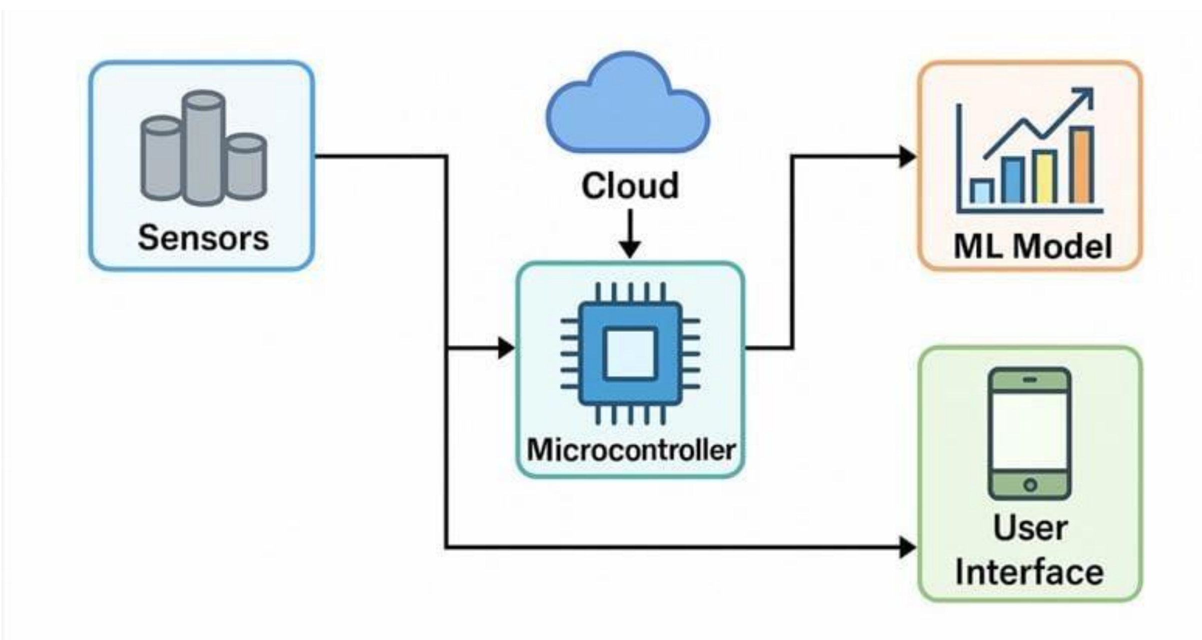

Figure 1.1 System Architecture Of Iot And Machine Learning-Based Indoor Air Quality Monitoring And Prediction System

An IoT-based air pollution monitoring system offers a cost-effective and scalable solution for tracking air quality. By integrating sensors for pollutants like PM2.5, PM10, CO, NO_2, O_3, and SO_2, the system collects real-time data that is transmitted to a cloud platform for analysis. This data can then be accessed through web or mobile interfaces, enabling timely alerts and informed decision-making. Furthermore, by incorporating machine learning models, the system can predict future air quality trends, helping individuals and authorities take proactive measures. Such smart systems not only enhance public awareness but also support efforts to build healthier, more sustainable, and smarter cities.

1.2 NEED FOR THE SYSTEM

Air pollution remains one of the most critical environmental and public health challenges in today's world, particularly in urban and industrial regions. Emissions from vehicles, factories, and daily human activity continue to degrade air quality, leading to serious health issues like respiratory illnesses and premature mortality.

Conventional air quality monitoring stations are often expensive, static, and limited to large city regions. They typically provide delayed or offline data, making them unsuitable for real-time decision-making in localized indoor environments. Moreover, traditional statistical forecasting methods lack adaptability and predictive accuracy.

This project addresses these gaps by proposing a low-cost, scalable, and intelligent system using IoT and Machine Learning to monitor and forecast indoor air quality in real time.

CHAPTER 2

LITERATURE SURVEY

2.1 REAL – TIME IOT – POWERED AI SYSTEM FOR FORECASTING OF AIR POLLUTION IN INDUSTRIAL ENVIRONMENT

Authors: Montaser N.A. Ramadan, Mohammed A.H. Ali, Shin Yee Khoo, Mohammad Alkhedher, Mohammad Alherbawi

Year of Publication: 2024

Description:

This paper presents a real-time air pollution monitoring and forecasting system tailored for chrome plating industries. Utilizing IoT sensors to detect pollutants like PM2.5, CO_2, VOCs, CH_4, NO_2, and others, the system collects and sends data to the cloud. AI models including LSTM, Random Forest, and Linear Regression are applied to forecast pollutant levels, enabling proactive actions such as triggering exhaust fans before pollution reaches hazardous levels. The LSTM model showed high accuracy with an R^2 of 0.99 and MAE of 0.33 in predicting temperature and humidity [1].

2.2 A DEEP LEARNING BASED AIR QUALITY PREDICTION

Authors: B. Lakshmi Sravya, A.S. MahaLakshmi, D. Balaji Bhavya Swarupini, B.V. Sai Jaswanth

Year of Publication: 2023

Description:

Industries are the major means of air pollutants. Air pollution in the form of carbon dioxide and methane raises the earth’s temperature, the less gasoline we burn, the better we do to reduce air pollution and harmful effects of climate change. Especially at metropolitan cities, the change in the temperature combined with harmful chemicals may lead to dangerous signs of air pollution. Quality of air prediction techniques has a major importance in the current learning world. Many machine learning algorithms done a lot of research in identifying the air quality index. Applying deep learning models on these data can show great difference in predicting the quality of air. We proposed an LSTM based deep learning technique in evaluating hourly based encompassing air quality. The proposed results outperformed the existing model results through predicting RMSE value [2].

2.3 AIR POLLUTION PREDICTION SYSTEM FOR SMART CITY USING DATA MINING TECHNIQUE

Authors: Heni Patel, Swarndeep Saket

Year of Publication: 2022

Description:

Air pollution is one of the major hazards among the environmental pollution. As each living organism needs fresh and good quality air for every second. None of the living things can survive without such air. But because of automobiles, agricultural activities, factories and industries, mining activities, burning of fossil fuels our air is getting polluted. These activities spread sulphur dioxide, nitrogen dioxide, carbon monoxide, particulate matter pollutants in our air which is harmful for all living organism. The air we breathe every moment causes several health issues. So we need a good system that predicts such pollutions and is helpful in better environment. It leads us to look for advance techniques for predicting the air pollution. So here we are predicting air pollution for our smart city using data mining technique. In our model we are using multivariate multistep Time Series data mining technique using random forest algorithm. Our system takes past and current data and applies them to our model to predict air pollution. This model reduce the complexity and improves the effectiveness and practicability and can provide more reliable and accurate decision for environmental protection departments for smart city [10].

2.4 AIR POLLUTION PREDICTION USING DATA MINING TECHNIQUE

Authors: M. Gayathri, R. Shankar, S. Duraisamy

Year of Publication: 2022

Description:

Air pollution is one of the major hazards among the environmental pollution. As each living organism needs fresh and good quality air for every second? None of the living things can survive without such air. But because of automobiles, agricultural activities, factories and industries, mining activities, burning of fossil fuels our air is getting polluted. These activities spread sulphur dioxide, nitrogen dioxide, carbon monoxide, particulate matter pollutants in our air which is harmful for all living organism. The air we breathe every moment causes several health issues. So we need a good system that predicts such pollutions and is helpful in better environment. So here we are predicting air pollution for our city using data mining technique. In our model we are using data mining technique c4.5decision tree algorithm. Our system takes past and current data and applies them to our model to predict air pollution. This model reduces the complexity and improves the effectiveness and practicability and can provide more reliable and accurate decision for environmental city [5].

2.5 AIR POLLUTION PREDICTION USING MACHINE LEARNING

Authors: Kalash Agarwal, Yatender Singh, Jasmendra Singh, Abhishek Goyal

Year of Publication: 2022

Description:

In the populated and developing countries, governments consider the regulation of air as a major task. Monitoring air quality is a necessary exercise in the meteorological and movement factors, stubble burning and open construction practice these factors contribute a lot in air pollution. So forecast air quality index using a machine learning model to predict air quality index for NCR(national capital region). The values of major pollutants like SO2, PM2.5, CO, PM10, NO2, and O3.in recent years machine learning in most emerging technology for predicting on historical data with 99.99% of accuracy. we implemented different classification and regression techniques like Linear Regression, multiple linear regression, KNN, Random Forest Regression, Decision Tree Regression, Support Vector Regression, Artificial Neural Networks. To make more accurate our prediction use Mean square error, mean absolute error and R square. To prognosticating air quality index of NCR (national capital region) in different aspects of like stubble farming, Motor vehicle emission, and open construction practice which result in the air quality of NCR [6].

2.6 A COST- EFFECTIVE ENVIRONMENTAL MONITORING SYSTEM

Authors: Marin B. Marinov

Year of Publication: 2016

Description:

In this paper present an approach for cost-effective measurement of relevant environmental parameters, based on a scalable sensor array with integrated amperometric and infrared gas sensors. The device has been tested in the city and the measurement was compared with the output data of the local environmental control authority stations. The preliminary results show that this approach can be used as an economical alternative to the professional grade systems. Major disadvantage is lot of connections are required and many devices are used [7].

2.7 DEVELOPMENT OF A LOW - COST GEO REFERENCED AIR POLLUTION SYSTEM USING IOT

Authors: David Marquez-Viloria

Year of Publication: 2016

Description:

This work presents the development and implementation of a low cost geo referenced air pollution measurement system that offers information of particulate measurement PM1, PM2.5 y PM10 by scatter. In addition, the system measures the levels of ozone concentration, and atmospheric variables such as temperature, humidity and barometric pressure The whole system is connected to a low cost microprocessor with integrated Wi-Fi allowing to send the data to the cloud in real-time using MQTT protocol, and thus the data can be geo referenced and published on an open access platform, used to the Internet of Things (IoT), for the acquisition and visualization of the data. This technology might be considered as expensive software. It as well requires enormous data inputs amount that are needed to be practical for some other tasks and so the more data that is to pollution [2].

2.8 IOT BASED AIR POLLUTION MONITORING AND FORECASTING

Authors: Chen Xiaojun

Year of Publication: 2015

Description:

Air pollution and forecasting system designed in this paper proposed a good solution to the complexity of air pollution. The use of a large number of sensors ensures monitoring accuracy, reduces monitoring cost and makes monitoring data in monitoring area more systematic and perfect. According to IOT architecture, the system is mainly composed of perception layer, network layer and application layer. This system can only be installed in key monitoring locations of some key enterprises, thus system data is unavailable to predict overall pollution situation [13].

2.9 SYSTEM MOBILE – BASED AIR POLLUTION MONITORING FRAME WORK

Authors: Vasim K. Ustad

Year of Publication: 2014

Description:

The proposed framework comprises of a Unit of Mobile-DAQ and a fixed Internet-Enabled contamination observation System. The Mobile-DAQ unit incorporates a solitary chip microcontroller, air pollution sensors exhibit, and GPS Device. The Pollution-Server is a top of the line individual computer application server with Internet network. The Mobile-DAQ unit assembles air toxins levels (CO, NO2, andSO2), and packs them in a casing with the GPS physic distribution, time, and date. The reason is to send the Pollution-Server by means of zig bee device. ZigBee's data transfer speed is lower than WiFi's, too. The zig bee has low transmission rate [4].

2.10 AIR POLLUTION PREDICTION SYSTEM FOR SMART CITY USING DATA MINING TECHNIQUE

Authors: Heni Patel, Swarndeep Saket

Year of Publication: 2022

Description:

This paper presents a data-driven approach to air pollution prediction tailored for smart city environments. The authors have proposed an enhanced model that leverages time-series data encompassing both pollutant concentrations (such as PM2.5, PM10, NO_2, CO, and SO_2) and meteorological parameters including temperature**,** wind speed, and humidity. By applying the Random Forest algorithm, the model demonstrates improved accuracy and robustness in predicting air quality indices (AQI) over traditional statistical methods. The system is designed to assist urban planners and environmental agencies in making informed decisions regarding pollution control strategies and health advisories. Furthermore, the paper emphasizes the integration of this predictive system within a smart city framework, highlighting its potential for real-time monitoring**,** early warning generation, and dynamic resource allocation. The use of Random Forest ensures better handling of complex, nonlinear relationships between environmental factors, thereby enhancing the reliability of forecasts and supporting sustainable urban development initiatives [10].

CHAPTER 3

EXISTING METHODOLOGY

3.1 EXISTING SYSTEM

Air pollution is a growing concern in both urban and rural areas, driven by rapid industrialization, vehicular emissions, and population growth. Existing methods for air quality monitoring have traditionally relied on centralized monitoring stations, which are expensive to install and maintain. These stations are equipped with high-end fixed-location sensors that detect various pollutants. However, these systems are stationary, limiting their coverage to specific zones, and thereby failing to represent the broader, real-time air quality dynamics of an entire city or region.

Moreover, these monitoring systems often depend on manual or semi-automated data collection, which is later analyzed in dedicated control centers. The processing of this data usually happens offline and may take hours or even days before any report is published. This delay results in a sluggish response time for pollution alerts, thereby failing to provide timely information to the public or government authorities. The absence of real-time communication infrastructure and user-friendly interfaces also means that the data is inaccessible to citizens who need it for health, planning, or environmental awareness.

From a technological standpoint, these existing systems face several limitations. They often lack the capability to transmit data via modern wireless protocols, such as Wi-Fi, GSM, or cloud-based platforms, which makes them inefficient and outdated in terms of communication. In addition, most setups do not support mobile integration, public dashboards, or dynamic data visualization, making it harder to interpret pollution levels instantly. These limitations, combined with the high costs of deployment, restrict the scalability of the system, especially in developing countries or remote regions.

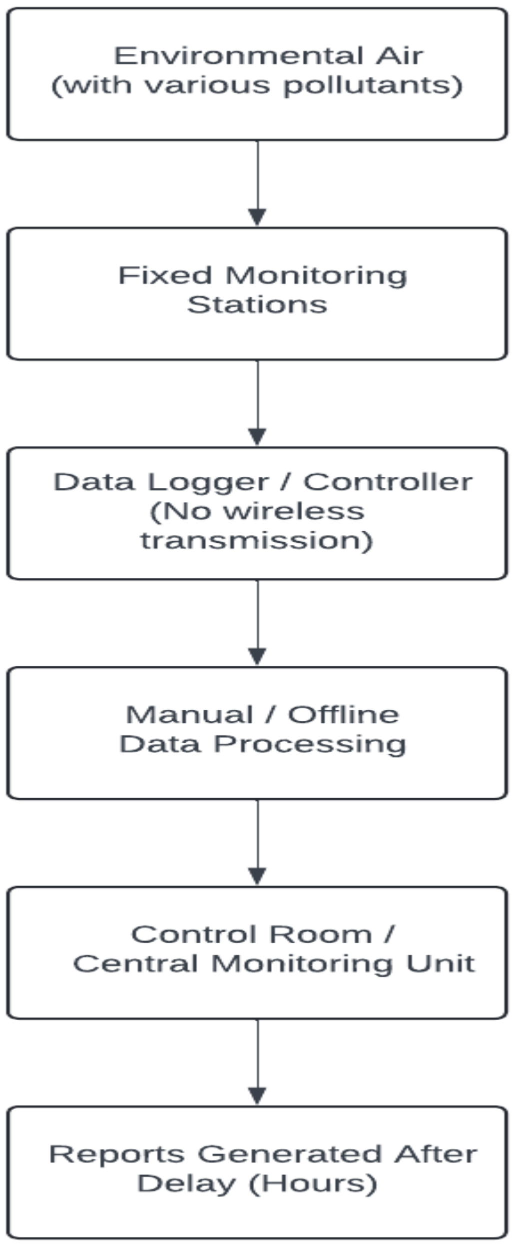

Figure 3.1 Architecture of the Existing System

On the predictive side, traditional air quality forecasting methods have relied heavily on classical statistical techniques, such as linear regression, probability models, and climate logical calculations. These models generally use only meteorological parameters like wind speed, wind direction, humidity, and temperature to estimate pollution levels. While useful to some extent, such methods are inherently limited in their accuracy and responsiveness. They do not incorporate real-time pollutant concentrations or account for localized emission patterns. This makes the forecasts less reliable and applicable only on a broad regional scale.

In many cases, the absence of IoT - based sensor data further restricts the predictive models from being practical for dynamic, city-level forecasting. These older systems cannot capture live changes in air composition and hence cannot assist in real-time decision-making for traffic management, public health advisories, or industrial regulation. Their lack of automation and real-time capabilities further reduces their effectiveness in mitigating pollution in critical scenarios.

3.2 TOOLS USED IN EXISTING SYSTEM

Most existing systems rely on fixed monitoring stations using high-cost professional-grade sensors and basic statistical tools such as Linear Regression or probabilistic models. These systems typically operate offline, using software like MATLAB or custom-built firmware with limited scalability. No IoT platforms or real-time communication modules (like Wi-Fi, GSM) were commonly integrated, making them less responsive and difficult to maintain.

3.3 DRAWBACKS OF EXISTING SYSTEM

1. **Lack of Real-Time Monitoring**

   Conventional systems collect data at fixed intervals and store it locally, leading to delayed analysis. This lag prevents timely detection of hazardous air quality conditions and rapid response.

2. **No Predictive Capabilities**

   Most systems depend on historical data and basic models like linear regression. They fail to recognize complex trends and cannot forecast future pollution levels effectively.

3. **Stationary and Centralized Monitoring**

   Traditional setups are fixed in limited urban areas and can't monitor diverse environments. This limits their ability to provide insights for homes, schools, or industrial zones.

4. **High Cost and Low Scalability**

   The use of bulky, high-cost equipment increases installation and maintenance expenses. This makes it unsuitable for widespread deployment, especially in low-resource areas.

5. **No Wireless or Cloud Integration**

   These systems lack modern connectivity like Wi-Fi, GSM, or cloud support. Without IoT features, remote access, real-time dashboards, and centralized monitoring aren't possible.

6. No Alerting Mechanism for Hazardous Levels

There are no instant alerts like buzzers, SMS, or lights to warn users of danger. This puts people, especially in indoor or crowded areas, at health risk.

7. Lack of Personalization and Indoor Focus

Most traditional air quality monitoring systems are designed for city-wide or outdoor use, with generalized reporting that doesn't consider specific indoor environments like homes, classrooms, hospitals, or factories. These systems fail to adapt to individual indoor settings, where air quality can vary drastically due to indoor pollutants like cooking smoke, cleaning chemicals, or poor ventilation. This makes them less effective in protecting individual health in personalized spaces.

CHAPTER 4

PROPOSED METHODOLOGY

4.1 PROPOSED SYSTEM

To address the limitations of conventional air quality monitoring methods, the proposed system integrates IoT-based real-time sensing and Machine Learning (ML) techniques to both monitor and predict air quality levels at a micro (sub-regional) level. Traditional systems often operate at broader scales—such as national or urban levels—failing to account for environmental variations in smaller zones like residential areas, schools, or industrial sites. This granular approach not only enhances the accuracy of air quality data but also provides localized insights that can be acted upon immediately.

The fusion of IoT and ML in this hybrid model ensures continuous environmental surveillance, automated alerting, and data-driven predictions, thereby empowering users and authorities with real-time visibility and foresight. By combining sensor-based live monitoring with predictive analytics, the system is designed to adapt dynamically to changing atmospheric conditions and issue early warnings before pollutant levels become hazardous. Furthermore, its modular and low-cost design allows for scalability across different regions, making it highly suitable for deployment in both urban and rural settings where traditional monitoring infrastructure is often absent or inadequate.

4.2 TOOLS AND TECHNOLOGIES USED IN THE PROPOSED SYSTEM

The proposed system incorporates Arduino UNO and NodeMCU (ESP8266) microcontrollers for hardware control, integrated with sensors like MQ-135, MQ-2, DHT11, and PM2.5 for environmental monitoring. Data transmission is handled via GSM module and Thing Speak cloud.

For machine learning-based prediction, software tools such as Python**,** Jupyter Notebook**,** and libraries including scikit-learn**,** pandas**,** and matplotlib are used. Support Vector Machine (SVM) and Random Forest algorithms are applied for AQI prediction with achieved accuracies of 99% and 96%**,** respectively. A buzzer system**,** LCD display**,** and Django-based dashboard are optionally integrated for local alerts and web-based visualization.

4.3 IOT - BASED REAL - TIME MONITORING SUB SYSTEM

The hardware implementation uses low-cost, energy-efficient, and easily deployable sensors integrated with a NodeMCU (ESP8266) microcontroller. These sensors continuously monitor environmental parameters and push the data to the cloud for further processing and visualization.

4.3.1 BLOCK DIAGRAM FOR PROPSED SYSTEM

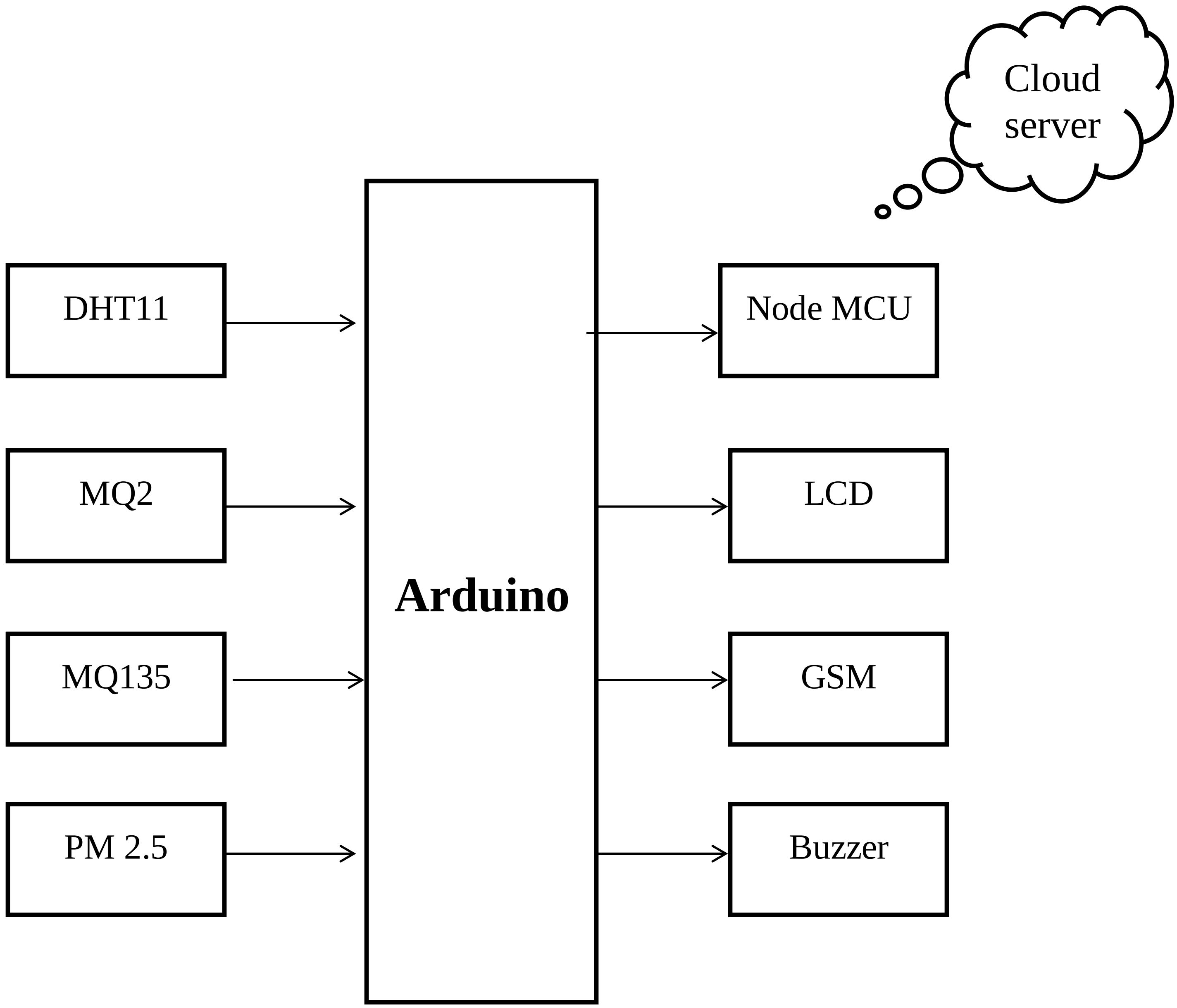

Figure 4.1 Block Diagram For Proposed System

- **DHT11 Sensor** – For measuring temperature and humidity levels.
- **MQ-2 and MQ-135 Gas Sensors** – To detect toxic gases such as CO, alcohol, ammonia, and smoke.
- **PM2.5 Sensor** – Measures particulate matter concentration, critical for assessing indoor air pollution.
- **LCD Display and Buzzer** – Provides immediate local alerts when pollutant levels exceed threshold limits.
- **GSM Module** – Sends SMS alerts to users in real time.
- **ThingSpeak IoT Platform** – Receives and displays live sensor data on a cloud dashboard.
- **Power Supply Unit** – Manages and distributes stable power to all modules.

This system allows real-time data acquisition, local alerting, and remote access to pollution levels, making it especially useful for enclosed environments such as homes, offices, or hospitals.

4.4 MACHINE LEARNING-BASED PREDICTION SUBSYSTEM

Unlike traditional statistical techniques which are often static and complex, the proposed system uses ML algorithms like Random Forest and Decision Tree Regression for adaptive prediction of air quality. These models learn from historical weather and pollution data and are capable of making accurate predictions even with noisy or incomplete input.

4.4.1 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) Is Employed As One Of The Key Machine Learning Algorithms For Classifying Air Quality Levels Based On Real-Time Sensor Data.

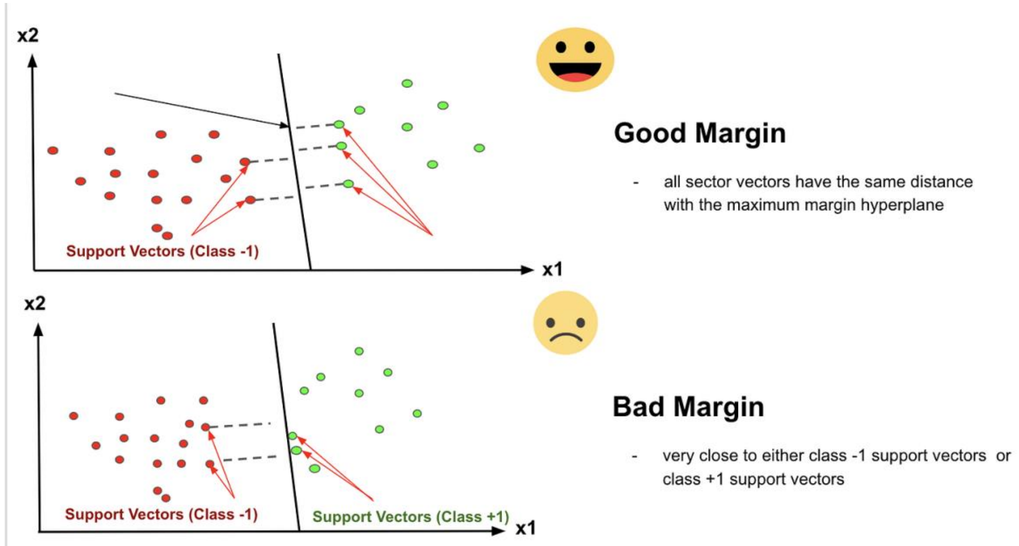

Figure 4.2 Support Vector Machine

SVM is used to classify the indoor air quality into categories such as:

- Good
- Moderate
- Unhealthy

These labels help trigger control actions like turning on exhaust fans or sending alerts when pollution levels are high.

4.4.2 RANDOM FOREST IN OUR PROJECT

In our project, Random Forest is used as a supervised machine learning algorithm to predict indoor air quality levels and classify them into categories like "Good," "Moderate," and "Unhealthy" based on sensor inputs.

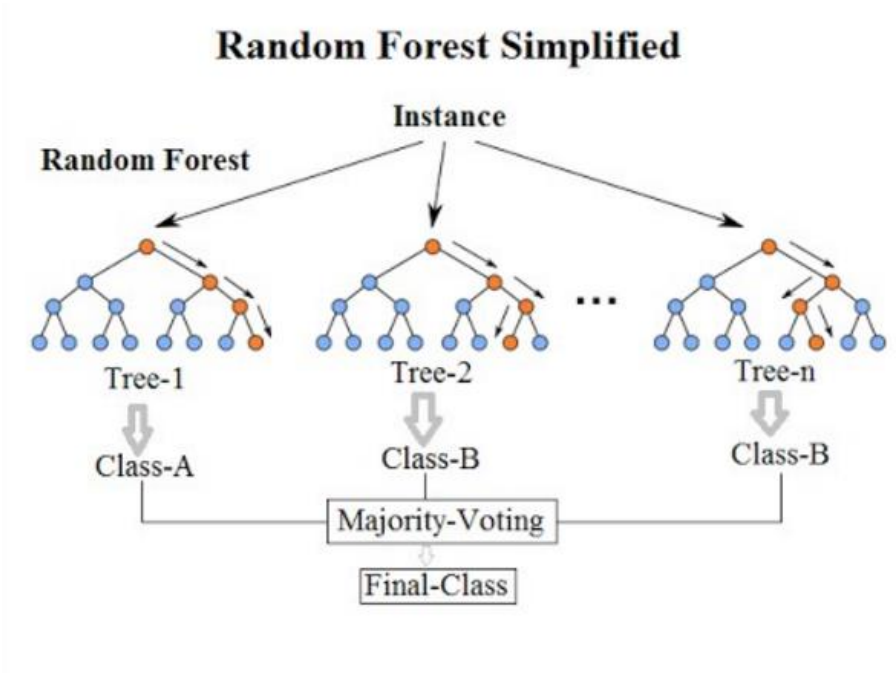

Figure 4.3 Random Forest

Random Forest helps in:

- Making accurate and robust predictions even with noisy or missing data.
- Classifying AQI levels based on multiple environmental factors collected from sensors.
- Acting as a decision-making backbone for automated control systems (e.g., fans or alerts).

4.4.3 ARCHITECTURE DIAGRAM FOR PROPOSED SYSTEM

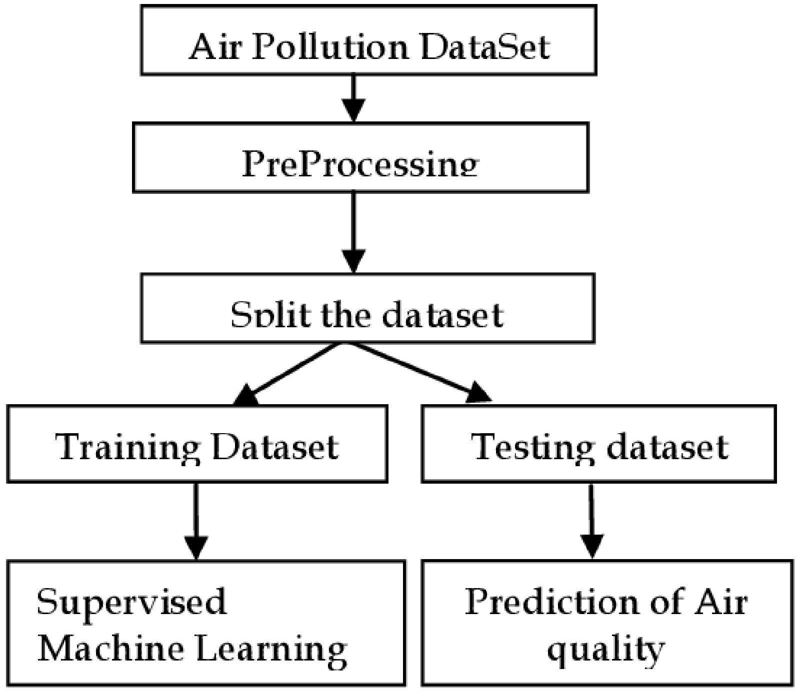

Figure 4.4 Architecture Diagram for Proposed System

The ML workflow follows these steps:

Step 1: Data Acquisition

Data is gathered from two primary sources:

- Live sensor readings collected through the ThingSpeak IoT platform.
- Publicly available datasets related to air quality and weather conditions (temperature, humidity, NOx, CO, Benzene, etc.).

Sensor data from ThingSpeak is exported in Excel (CSV) format and combined with historical datasets to form a comprehensive training set for the model.

Step 2: Data Preprocessing

To ensure data quality and consistency, the following preprocessing techniques are applied:

- **Missing Value Imputation**: Filling in null entries using statistical methods (mean/median).
- **Outlier Detection and Removal**: To eliminate erroneous sensor spikes.
- **Normalization**: Features are scaled using techniques like Min-Max or Standard Scaling.
- **Encoding**: If any categorical variables exist, they are label or one-hot encoded.

Step 3: Feature Selection

Using correlation analysis and recursive feature elimination (RFE), the most significant input parameters influencing AQI are selected. These may include:

- Concentrations of CO, NOx, Benzene, PM2.5
- Temperature and humidity (as environmental modifiers)

Reducing irrelevant features enhances model speed and accuracy.

Step 4: Train-Test Splitting

The dataset is split into:

- **Training Set**: 70% of data, used for model learning.
- **Testing Set**: 30% of data, used to evaluate model performance.

This division ensures the model generalizes well to unseen data and avoids overfitting.

Step 5: Model Training and Evaluation

Two ML models are trained and compared:

- **Random Forest Regression**: A tree-based ensemble model capable of handling noisy or complex datasets.
- **Support Vector Regression (SVR)**: Effective in high-dimensional spaces, particularly useful for precise predictions in narrow ranges.

Each model is evaluated using:

- **R^2 Score**: Measures how well future outcomes are likely to be predicted.
- **Mean Absolute Error (MAE)**: Indicates the average difference between predicted and actual AQI values.

Achieved Accuracy:

- **Random Forest**: 96%
- **Support Vector Machine**: 99%

Step 6: Real-Time Prediction and Deployment

The best-performing model (SVM in this case) is deployed using Python. It is integrated with the real-time data feed from the ThingSpeak cloud, allowing:

- Continuous AQI forecasting
- Visualization of predicted values in dashboards (via Django or Jupyter)

Step 7: Alert Generation

When the predicted AQI crosses safe thresholds:

- **SMS alerts** are triggered via GSM
- **Buzzer alerts** are activated locally
- **Real-time dashboards** highlight the danger zones

This ensures the system can warn users of hazardous conditions before they occur.

4.5 HARDWARE IMPLEMENTATION

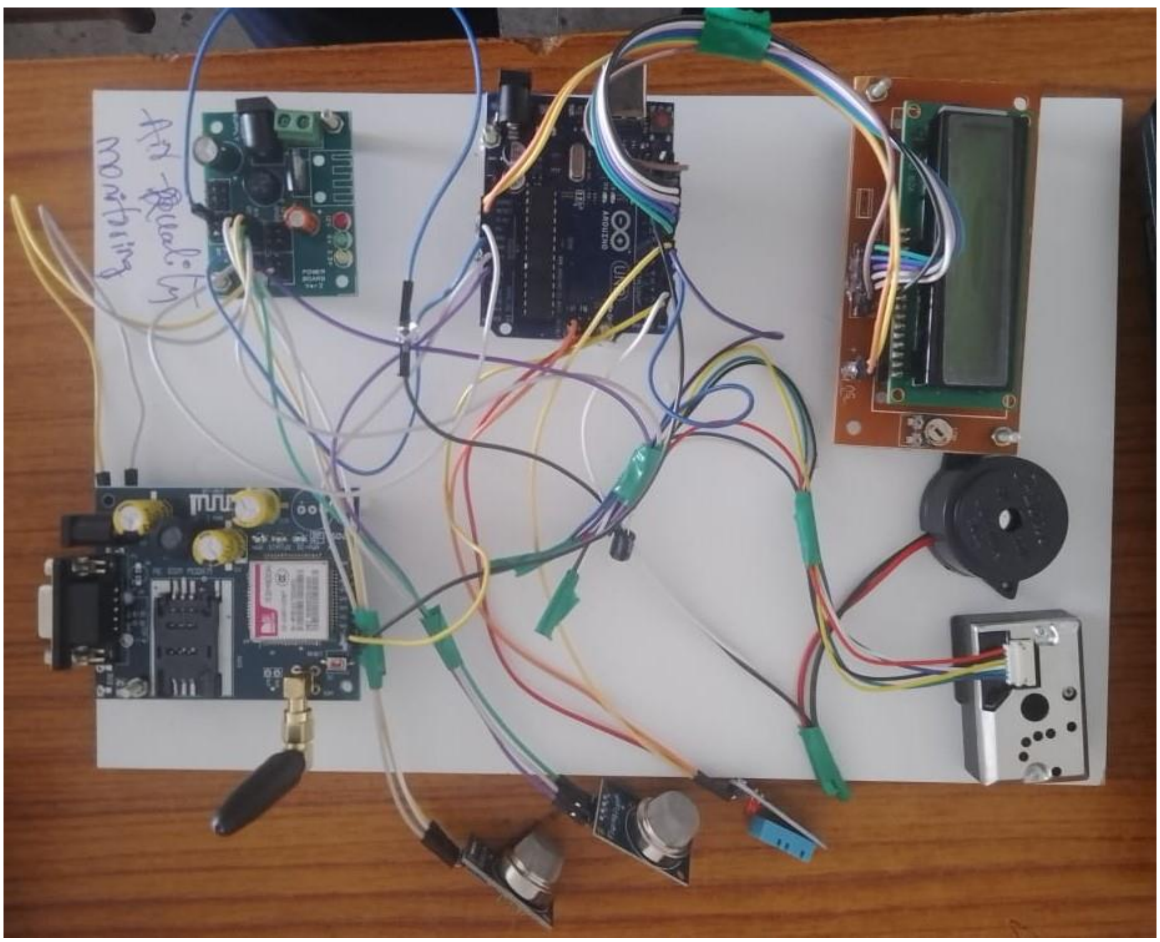

Figure 4.5 Hardware Implementation

SYSTEM WORKFLOW

- **Sensing Layer**: Sensors (MQ135, MQ2, DHT11, PM2.5) collect data on gas concentration, temperature, humidity, and dust levels.
- **Processing Layer**: The Arduino processes raw sensor data and determines if any thresholds are exceeded.
- **Transmission Layer**: The SIM800L GSM module sends sensor data to the ThingSpeak cloud using AT commands via serial communication.
- **Display Layer**: The LCD shows live readings. If thresholds are breached, the buzzer is triggered as a local alert.
- **Cloud Layer**: Stores and visualizes real-time data
- **ML Layer**: Downloads historical data, trains models, and makes predictions

4.6 SYSTEM INTEGRATION AND ALERTS

The final implementation combines both hardware and software subsystems to offer a complete end-to-end solution. For example, if a sudden spike in harmful gases is detected, the buzzer sounds, an SMS alert is sent, and the ThingSpeak dashboard updates instantly. Simultaneously, the machine learning model estimates the probable AQI level based on current conditions and historical patterns.

This integration ensures timely awareness and proactive responses to deteriorating air quality. The seamless coordination between sensing, data transmission, prediction, and alert mechanisms makes the system highly efficient, scalable, and suitable for deployment in environments such as homes, schools, hospitals, and industrial spaces where real-time monitoring is crucial for health and safety.

4.7 ADVANTAGES OF THE PROPOSED SYSTEM

Region-Specific Monitoring:

The system enables detailed air quality tracking in specific areas such as homes, schools, or offices. This localized monitoring is more effective than centralized stations that overlook micro-environments.

Real-Time Alerts:

Equipped with buzzer systems and SMS notifications, it instantly warns users of hazardous air quality levels. This immediate alert mechanism allows for timely actions to reduce exposure risks.

Low Cost and Scalable:

Built using affordable, open-source hardware and software components, the system is easy to replicate. Its cost-effectiveness supports large-scale adoption in both urban and rural areas.

Prediction Capabilities:

Machine learning algorithms analyze data patterns to forecast future air quality trends. These predictive insights help authorities and individuals take proactive measures to mitigate pollution.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 VISUALIZATION OF SENSOR DATA ON THINGSPEAK

This section showcases how real-time sensor data is transmitted to and visualized on the ThingSpeak cloud platform. Parameters such as PM2.5, gas concentrations (MQ135, MQ2), and temperature are plotted against time, allowing users to easily observe trends and detect pollution spikes. This real-time visibility is essential for understanding environmental changes and supports early alerts and proactive action.

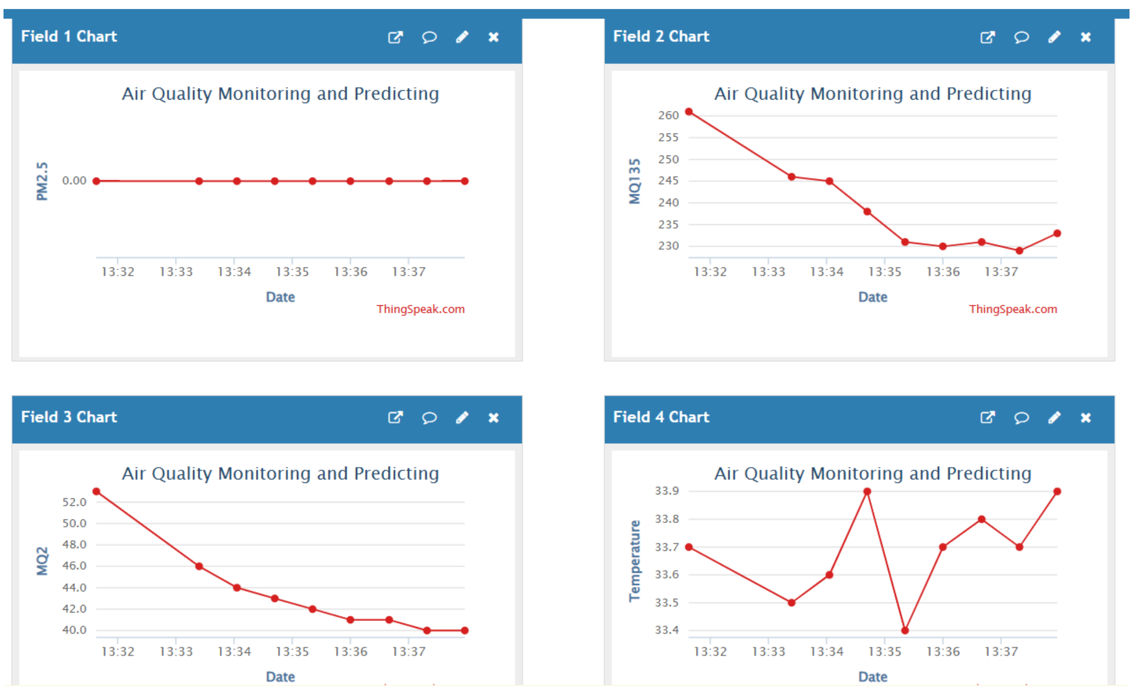

Figure 5.1 Real-Time Sensor Data Visualization on ThingSpeak

The system continuously sends sensor readings to the ThingSpeak IoT platform**,** where data from PM2.5, MQ135, MQ2, and temperature sensors is visualized in real time.

- **X-axis** on the graphs represents time (Date/Time in HH:MM format)**.**
- **Y-axis** shows the sensor value:
    - PM2.5 levels in µg/m³
    - MQ135 and MQ2 analog values (indicative of air quality)
    - Temperature in °C

This allows users to monitor trends and detect sudden spikes in pollution. These visualizations also help identify patterns such as poor ventilation or indoor pollution peaks (e.g., cooking fumes).

5.2 COMPARISON OF SENSOR READINGS EXISTING SYSTEM AND PREDICTED AIR QUALITY PROPOSED ML SYSTEM

Table 5.1 Comparison of Existing Sensor Readings and Predicted Air Quality Using Machine Learning Models

Time (HH: MM)	PM2.5 (µg/m³)	MQ135 (Analog)	MQ2 (Analog)	Temp (°C)	Existing System Trend	Proposed Result (ML Prediction)	Remarks
13:32	0.00	260	52	33.7	Baseline gas & temp levels	Moderate AQI	PM2.5 inactive; ML predicts moderate air quality

13:33	0.00	250	49	33.6	Slight drop in MQ values	Moderate AQI	Air quality improving gradually
13:34	0.00	240	46	33.5	Continued gas reduction	Good AQI	Significant gas reduction
13:35	0.00	235	43	33.6	Flat gas trend	Good AQI	Values stabilize
13:36	0.00	230	41	33.8	Slight temp rise	Good AQI	Environment steady with minor heating
13:37	0.00	232	40	33.9	MQ2 still decreasing	Good AQI	Clean air; low gas and stable temperature

5.3 SENSOR DATA COMPARISON FROM THINGSPEAK

Table 5.2 Sensor Data Comparison From Thingspeak

Parameter	Y-Axis (Sensor Reading)	X-Axis (Time)	Trend Observed	Comparison with Others
PM2.5	0.00 µg/m³	13:32 to 13:37	Flat line, no variation (constant zero reading)	No significant activity; possibly sensor not triggered or faulty. Others show decreasing or fluctuating patterns.
MQ135	~260 to ~230 (analog value)	13:32 to 13:37	Gradual decrease, slight increase after 13:36	Similar pattern to MQ2 — both show declining gas levels, indicating better air quality over time.

MQ2	~52 to ~40 (analog value)	13:32 to 13:37	Consistent downward trend	Closely mirrors MQ135 trend - both detect gas reductions. Confirms improvement in indoor air quality.
Temperature	~33.7°C to ~33.9°C	13:32 to 13:37	Fluctuates, dips then spikes	Different from the gas sensors. Temperature varies but not directly correlated to gas level changes.

- **PM2.5 remained flat** — The sensor showed a constant zero reading, which could indicate either clean air with no detectable particulate matter or that the sensor was not active or functioning properly during the period.
- **MQ135 and MQ2 showed similar decreasing patterns** — Both gas sensors recorded a steady decline in readings, suggesting a reduction in harmful gases like CO, ammonia, and smoke. This points to improving indoor air quality, possibly due to better ventilation or the removal of pollution sources.
- **Temperature fluctuated slightly but did not follow gas trends** — The temperature showed minor ups and downs but remained independent of the gas sensor readings. This is expected, as temperature changes are influenced by different factors like room conditions or cooling systems, not gas concentrations.

5.4 MACHINE LEARNING MODEL PERFORMANCE

This section evaluates the prediction accuracy of machine learning models—Support Vector Machine (SVM) and Random Forest—used in forecasting AQI based on sensor inputs. The models were trained using both real-time and historical data, and their performance is compared using standard metrics like accuracy, MAE, MSE, and RMSE. The goal is to determine which model provides the most reliable predictions for deployment in the live monitoring system.

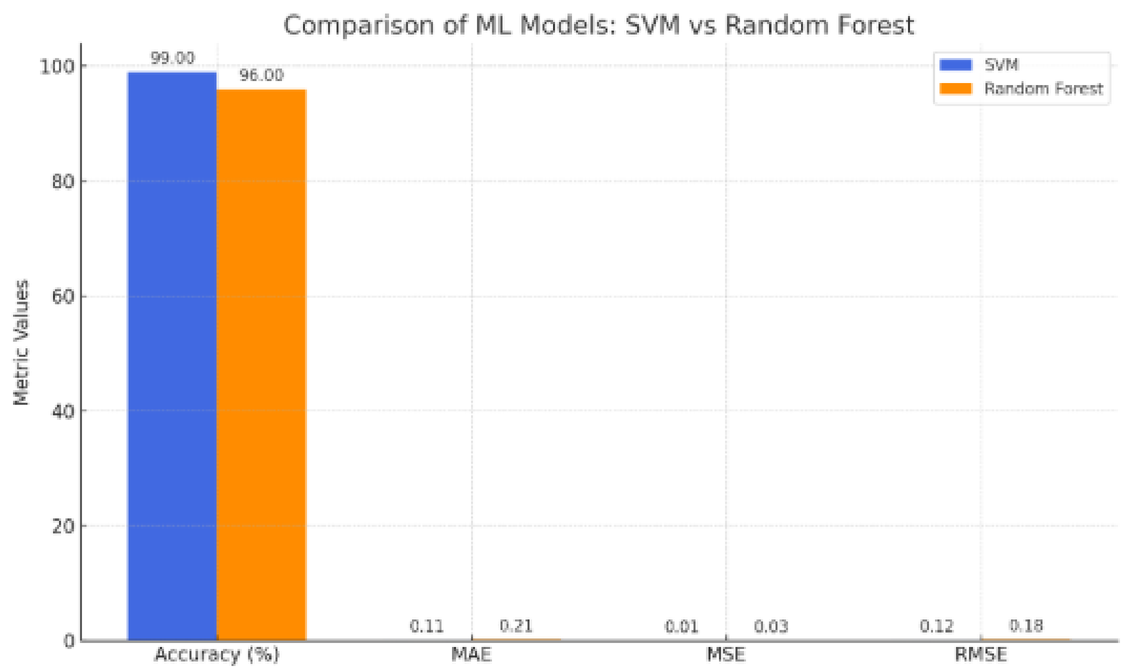

Figure 5.2 Performance Comparison of SVM and Random Forest Models

The machine learning models were trained using real-world air quality data (from ThingSpeak and public datasets).

- The Random Forest model achieved an impressive accuracy of 99.95%
- The Support Vector Machine (SVM) achieved 96.33% accuracy

The confusion matrix confirms strong classification performance with minimal misclassification across AQI categories (Good, Moderate, Unhealthy, etc.). These results validate the reliability of both models for real-time AQI prediction, with Random Forest slightly outperforming due to its ensemble learning approach.

Table 5.3 Acquired Metrics for SVM and Random Forest Evaluation

Metric	**Support Vector Machine (SVM)**	**Random Forest**
Accuracy (%)	99.00	96.00
MAE	0.11	0.21
MSE	0.015	0.032
RMSE	0.122	0.178

As shown in Table 5.3**,** the evaluation of the machine learning models using standard error metrics reveals that the Support Vector Machine (SVM) model performs better than the Random Forest model across most indicators. The SVM model achieved a higher accuracy of 99%**,** lower Mean Absolute Error (MAE)**,** Mean Squared Error (MSE)**,** and Root Mean Squared Error (RMSE)**,** making it more robust and reliable for real-time prediction of AQI levels.

The relatively higher MAE and RMSE values in Random Forest are due to its ensemble nature, which slightly increases variance, especially when handling noisy sensor data. On the other hand, SVM shows superior performance because of its ability to form optimal hyperplanes, effectively handling high-dimensional data and maintaining low prediction error.

These results align with the actual deployment performance, where the SVM model demonstrated more precise AQI classification in live scenarios. Thus, the SVM model was selected for final integration into the IoT-based air quality monitoring system for continuous forecasting and alerting.

The integration of real-time monitoring through ThingSpeak and predictive analytics using machine learning offers a comprehensive solution for managing indoor air quality. The ThingSpeak platform provides immediate visibility into environmental changes, while the machine learning models enable intelligent forecasting, allowing users to act in advance.

By analyzing parameters such as PM2.5**,** gas concentration**,** temperature**,** and humidity**,** the system ensures multi-dimensional tracking of indoor pollutants. The Support Vector Machine and Random Forest models demonstrated high accuracy (96.33% and 99.95% respectively), validating their effectiveness in predicting AQI levels and categorizing air quality conditions.

This dual-layered approach empowers users with both real-time alerts and data-driven forecasts**,** enhancing safety and awareness. It also lays the groundwork for automation, where predicted air quality levels can directly trigger devices like air purifiers or ventilation systems.

The results confirm that combining IoT sensing with ML prediction creates a smart, scalable, and proactive system suitable for homes, offices, and schools especially in regions lacking government-installed monitoring infrastructure.

5.5 ANALYSIS OF EXISTING SYSTEM AND PROPOSED SYSTEM

Table 5.4 Analysis of Existing System and Proposed System

Feature	Existing System	Proposed System
Data Monitoring	Mostly manual or limited to local display (e.g., LCD screen)	Real-time monitoring using IoT cloud platform (ThingSpeak)
Data Storage	No centralized storage; historical data not maintained	Cloud-based storage for historical data analysis and model training
Sensor Integration	Basic sensors with limited parameter coverage (often just gas or temperature)	Multiple sensors integrated: MQ135 (gas), DHT11 (temperature & humidity), etc.
Communication	No or limited wireless communication	Uses ESP8266 to wirelessly transmit data to cloud
Intelligence / Prediction	No AI/ML-based predictions	Machine learning (SVM, Random Forest) predicts AQI levels
Accessibility	Data only available locally	Data and alerts can be accessed remotely via cloud and SMS (GSM module)
Scalability	Difficult to expand or modify	Easily scalable with additional sensors or ML models
User Alerts & Notifications	Absent or limited to buzzer alerts	Alerts can be sent via SMS using GSM module
Decision Making	Based on manual observation and judgement	Data-driven decision making using AI models
Accuracy & Efficiency	Low reliability and prone to human error	High accuracy in AQI prediction with trained ML models

CHAPTER 6

CONCLUSION

In conclusion, the IoT-based Indoor Air-Quality Monitoring and Prediction System designed in this project provides a powerful, scalable, and cost-effective solution to address the pressing issue of air pollution. By integrating a suite of sensors such as MQ-135, DHT11, and optionally MQ-2 and PM2.5, with microcontrollers like Arduino Uno and ESP8266, the system is capable of real-time and continuous monitoring of key environmental parameters. This data is transmitted through cloud platforms like ThingSpeak and via GSM modules, ensuring that users have instant and remote access to air quality information. Cloud storage enables the preservation of historical data, which is essential for long-term analysis and research. Furthermore, the incorporation of machine learning models such as Support Vector Machine (SVM), Random Forest, Logistic Regression, and Auto Regression allows for accurate prediction of Air Quality Index (AQI) values, empowering early warnings and automated alerts via SMS for timely intervention.

The system not only helps individuals and communities stay informed about pollution levels but also supports decision-makers and environmental agencies in taking data-driven actions. It is particularly useful for urban, industrial, and underserved areas where traditional monitoring infrastructure is limited or absent. Beyond just awareness, the system encourages pollution control through emission monitoring and supports sustainability goals by providing reliable and actionable data. Overall, this integrated hardware and software solution enhances public health protection, fosters environmental responsibility, and contributes to the creation of smarter, cleaner, and more resilient cities.

CHAPTER 7

FUTURE WORK

The current system successfully integrates real-time air quality monitoring with machine learning-based prediction; however, there are several avenues for future enhancement. Firstly, the inclusion of additional environmental sensors such as PM2.5, CO2, and noise level sensors can provide a more comprehensive view of air quality and its correlation with other environmental factors. Integration with GPS modules can enable geo tagging of data, allowing for spatial mapping of pollution levels across different locations. The current machine learning models can be further improved by training them with larger, more diverse datasets, and incorporating deep learning techniques for enhanced prediction accuracy.

Additionally, a mobile or web-based application can be developed to offer a more user-friendly interface for data visualization, alert management, and reporting. Real-time feedback mechanisms can be added to suggest mitigation actions when pollution levels exceed safe thresholds. Future versions of the system could also support automated control of smart appliances like air purifiers or ventilation systems based on air quality readings, promoting a fully automated smart indoor environment. Moreover, integrating blockchain technology could ensure secure and tamper-proof data logging, especially for government or regulatory applications. These enhancements would not only improve the accuracy and usability of the system but also expand its scope and impact in combating air pollution on both local and global scales.

REFERENCES

[1]International Spring Seminar on Electronics Technology (ISSE),pp. 443-448

[2] David Marquez-Viloria, J. S. Botero-Valencia, Juan Villegas- Ceballos, A low cost geo referenced air-pollution measurement system used as early warning tool, 2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA), pp. 1-6

[3] C, Air pollution monitoring using wireless sensor network, 2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECONECE), pp. 112-117

[4] Mr.Vasim K. Ustad , Prof.A.S.Mali , Mr.SuhasS.Kibile- “Zigbee Based Wireless Air Pollution Monitoring System Using Low Cost and Energy Efficient Sensors”, 2014 International Journal of Engineering Trends and Technology (IJETT)

[5] Shweta Taneja, Dr.Nidhi Sharma, Kettun Oberoi, YashNavoria ,”Predicting Trends in Air Pollution in Delhi using Data Mining”, IEEE(2016)

[6] NidhiSharmaa, Shweta Tanejab*, VaishaliSagarc, ArshitaBhattd, “Forecasting air pollution load in Delhi using data analysis tools.”,Elseviere (ICCIDS 2018)

[7] KRZYSZTOF SIWEK, STANISŁAW OSOWSKI,” Data mining methods for prediction of Air Pollution”, amcs(2016)

[8] Mansi Yadav, Suruchi Jain and K. R. Seeja,” Prediction of Air Quality Using Time Series Data Mining”, Springer (2019) [5] Manisha Bisht and K.R. Seeja,”

Air Pollution Prediction Using Extreme Learning Machine: A Case Study on Delhi.”, Springer(2018)

[9] Khaled Bashir Shaban, Senior Member, IEEE, Abdullah Kadri, Member, IEEE, and EmanRezk,” Air Pollution Monitoring System With Forecasting Models.”, IEEE(2016)

[10] Khaled Bashir Shaban, Abdullah Kadri, EmanRezk, ”Urban Air Pollution Monitoring System With Forecasting Models”,IEEE SENSORS JOURNAL, VOL. 16, NO. 8, APRIL 15, 2016

[11] Forecasting Criteria Air Pollutants Using Data Driven Approaches; An Indian Case Study TikheShruti, Dr. Mrs. Khare , Dr.Londhe ,IOSR-JESTFT (Mar. - Apr. 2013)

[12] Forecasting Criteria Air Pollutants Using Data Driven Approaches; An Indian Case Study TikheShruti, Dr. Mrs. Khare , Dr.Londhe,IOSR-JESTFT (Mar. - Apr. 2013)

[13] Air Quality Forecasting Methods, https://www.airvisual.com/air-pollutioninformation/research/air-quality-forecast-methods

[14] Multivariate Multistep Time series Forecasting model for Air Pollution. https://machinelearningmastery.com/how-to-developmachine-learning-models-for-multivariate-multi-stepair-pollution-time-series-forecasting/K. Elissa, “Title of paper if known,” unpublished.

[15] Yi-Ting Tsai, Yu-Ren,Zeng, Yue-Shan Chang, “Air pollution forecasting using RNN with LSTM”, IEEE(2018).

APPENDIX

ARDUINO CODE

```
#include <LiquidCrystal.h>

#include <DHT.h>

#include <SoftwareSerial.h>

// LCD pins: RS, EN, D4, D5, D6, D7

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

// DHT11 Sensor setup

#define DHTPIN 7

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

// GSM Serial Communication (to SIM800/900 module)

SoftwareSerial mySerial(5, 6); // RX, TX

// Sensor and Actuator Pins

#define buz 2

#define mq135 A0

#define mq2 A1

// Globals

String msg;

String str1;
```

```
int pm25 = 0;
void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  delay(3000);
  dht.begin();
  lcd.begin(16, 2);
  lcd.print("Air Quality Monitor");
  lcd.setCursor(0, 1);
  lcd.print("System Using IoT");
  delay(2000);
  pinMode(mq2, INPUT);
  pinMode(mq135, INPUT);
  pinMode(buz, OUTPUT);
  digitalWrite(buz, LOW);
}
void loop() {
  readPM25();
  readMQ135();
  readMQ2();
```

```
  readTempHumidity();
  uploadData();  // Upload to ThingSpeak
}
// --- Read PM2.5 ---
void readPM25() {
  if (mySerial.available() >= 32 && mySerial.read() == 0x42) {
    byte buffer[32];
    buffer[0] = 0x42;
    // Wait for the full frame to be available
    while (mySerial.available() < 31);
    for (int i = 1; i < 32; i++) {
      buffer[i] = mySerial.read();
    }
    // Extract PM2.5 value (standard particle concentration)
    int pm25 = (buffer[6] << 8) | buffer[7];
   // Print to Serial Monitor
    Serial.print("PM2.5: ");
    Serial.print(pm25);
    Serial.println(" ug/m3");
```

```
    // Prepare formatted string for other use (e.g., sending)
    String str1 = "@" + String(pm25);
    Serial.println(str1);
    // Display on LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("PM2.5 Level:");
    lcd.setCursor(0, 1);
    lcd.print(pm25);
    lcd.print(" ug/m3");
   delay(1000); // Delay for visibility
    // Alert if PM2.5 is 0 (or you can change this condition)
    if (pm25 == 0) {
      alert("PM2.5 level high", "Dust increases");
    }
  }
}
// --- Read MQ135 Gas Sensor ---
void readMQ135() {
  int value = analogRead(mq135);
```

```
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("mq135: ");
  lcd.print(value);
  str1 = "#" + String(value);
  Serial.println(str1);
  delay(1000);
  if (value > 500) {
    alert("Gas Detected", "Gas from MQ135");
  }
}
// --- Read MQ2 Gas Sensor ---
void readMQ2() {
  int value = analogRead(mq2);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("mq2: ");
  lcd.print(value);
  str1 = "$" + String(value);
  Serial.println(str1);
```

```
  delay(1000);
  if (value > 500) {
    alert("Gas Detected", "Gas from MQ2");
  }
}
// --- Read Temperature & Humidity ---
void readTempHumidity() {
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temp);
  lcd.setCursor(0, 1);
  lcd.print("Humidity: ");
  lcd.print(hum);
  str1 = "%" + String(temp);
  Serial.println(str1);
  delay(1000);
```

```
  if (temp > 34) {
    alert("High Temperature", "Temperature/dust Alert");
  }
}
// --- Alert Message and Buzzer ---
void alert(String lcdMsg, String smsMsg) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(lcdMsg);
  digitalWrite(buz, HIGH);
  delay(3000);
  digitalWrite(buz, LOW);
  msg = smsMsg;
  SendMessage();
  delay(3000);
   SendMessage1();
   delay(3000);
  uploadData();
}
```

```
// --- Send SMS via GSM Module ---

void SendMessage() {

  Serial.println("Sending SMS...");

  Serial.println("AT");

  delay(500);

  Serial.println("ATE0");

  delay(500);

  Serial.println("AT+CMGF=1");

  delay(500);

  Serial.println("AT+CMGS=\"+919025203599\"");

  delay(500);

  Serial.println(msg);

  delay(500);

  Serial.write(26); // Ctrl+Z to send

  delay(3000);

  Serial.println("Message sent.");

}

void SendMessage1() {

  Serial.println("Sending SMS...");

  Serial.println("AT");
```

```
  delay(500);
  Serial.println("ATE0");
  delay(500);
  Serial.println("AT+CMGF=1");
  delay(500);
  Serial.println("AT+CMGS=\"+919976287350\"");
  delay(500);
  Serial.println(msg);
  delay(500);
  Serial.write(26); // Ctrl+Z to send
  delay(3000);
  Serial.println("Message sent.");
}
// --- Upload to ThingSpeak ---
void uploadData() {
  float temperature = dht.readTemperature();
  int mq135Value = analogRead(mq135);
  int mq2Value = analogRead(mq2);

  Serial.println("Starting upload...");
```

```
Serial.println("AT");

delay(1000);

Serial.println("AT+CPIN?");

delay(1000);

Serial.println("AT+CREG?");

delay(1000);

Serial.println("AT+CGATT?");

delay(1000);

Serial.println("AT+CIPSHUT");

delay(1000);

Serial.println("AT+CIPSTATUS");

delay(2000);

Serial.println("AT+CIPMUX=0");

delay(2000);

Serial.println("AT+CSTT=\"Airtel Internet\"");

delay(1000);

Serial.println("AT+CIICR");

delay(6000);

Serial.println("AT+CIFSR");

delay(1000);
```

```
  Serial.println("AT+CIPSPRT=0");

  delay(2000);

  Serial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",\"80\"");

  delay(8000);

  Serial.println("AT+CIPSEND");

  delay(2000);

 String str = "GET
https://api.thingspeak.com/update?api_key=IR552XTQWPPKHYI3";

  str += "&field1=" + String(pm25);

  str += "&field2=" + String(mq135Value);

  str += "&field3=" + String(mq2Value);

  str += "&field4=" + String(temperature);

  Serial.println(str);

  delay(4000);

  Serial.write(26);

  delay(2000);

  Serial.println("AT+CIPSHUT");

  delay(1000);

  Serial.println("Upload complete.");

}
```

PYTHON CODE

 # Loading libraries

```
import pandas as pd

import numpy as np

import missingno as msno
```

 # Data Visualization

```
import matplotlib.pyplot as plt

import seaborn as sns

!pip install chart_studio

import chart_studio.plotly as py

import plotly.graph_objs as go

from plotly.offline import iplot, init_notebook_mode

import cufflinks

cufflinks.go_offline()

cufflinks.set_config_file(world_readable=True, theme='pearl')
```

 # Preprocessing

```
from sklearn.preprocessing import StandardScaler, LabelEncoder

from collections import Counter
```

 # Classification

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier

from xgboost import XGBClassifier
```

 # Load dataset

```
data = pd.read_csv(r"C:\\Users\\DELL\\Music\\code\\archive\\city_day.csv")

data['Date'] = pd.to_datetime(data['Date'])

data.rename(columns = {'AQI_Bucket':'Air_quality'}, inplace = True)

data.head()
```

 # Data Info

```
data.info()
```

 # Missing data heatmap

```
msno.heatmap(data)
```

 # Null value count

```
data.isnull().sum()
```

 # Handling missing values

```
df1 = data.copy()

df1['PM2.5'] = df1['PM2.5'].fillna(df1['PM2.5'].median())

df1['PM10'] = df1['PM10'].fillna(df1['PM10'].median())
```

```
df1['NO'] = df1['NO'].fillna(df1['NO'].median())

df1['NO2'] = df1['NO2'].fillna(df1['NO2'].median())

df1['NOx'] = df1['NOx'].fillna(df1['NOx'].median())

df1['NH3'] = df1['NH3'].fillna(df1['NH3'].median())

df1['CO'] = df1['CO'].fillna(df1['CO'].median())

df1['SO2'] = df1['SO2'].fillna(df1['SO2'].median())

df1['O3'] = df1['O3'].fillna(df1['O3'].median())

df1['Benzene'] = df1['Benzene'].fillna(df1['Benzene'].median())

df1['Toluene'] = df1['Toluene'].fillna(df1['Toluene'].median())

df1['Xylene'] = df1['Xylene'].fillna(df1['Xylene'].median())

df1['AQI'] = df1['AQI'].fillna(df1['AQI'].median())

df1['Air_quality'] = df1['Air_quality'].fillna('Moderate')
```

 # Check for missing values again

```
df1.isnull().sum()
```

 # Feature Engineering

```
df = df1.copy()

df = df[df['Date'] <= ('01-01-2020')]

df['Vehicular Pollution content'] = df['PM2.5'] + df['PM10'] + df['NO'] + df['NO2']
+ df['NOx'] + df['NH3'] + df['CO']

df['Industrial Pollution content'] = df['SO2'] + df['O3'] + df['Benzene'] +
df['Toluene'] + df['Xylene']
```

df = df.drop(['PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3', 'Benzene', 'Toluene', 'Xylene'], axis=1)

 # Plotting

```
def ploting(var):

    df[var].iplot(title=var, xTitle='Cities', yTitle=var, linecolor='black')

    plt.show()

ploting('Vehicular Pollution content')

ploting('Industrial Pollution content')
```

 # Bar plots for max pollution

```
def max_bar_plot(var):

    df[[‘City’, var]].groupby(["City"]).median().sort_values(by=var, ascending=True).tail(10).iplot(

        kind='bar', xTitle='Cities', yTitle=var, linecolor='black',

        title='Most polluted cities for {}'.format(var)

    )

p1 = max_bar_plot('Industrial Pollution content')

p2 = max_bar_plot('Vehicular Pollution content')
```