Air Quality Monitor and Prediction System Using Iot and Machine Learning Algorithms

[a]E.Sasikala,[b]Koppala Komala, [c]Perimidi Sirisha, [d]Dhara Kishore,[e]Pigilam Sandhya, [f]Pulla Yedukondalu.

Assistant Professor, Department of ECE, Gokula Krishna College of Engineering Jawaharlal Nehru Technological University Anantapur.

Abstract: **The pollution level is increasing rapidly due to factors like industries, urbanization, increase in population, and vehicle use which can affect human health. IOT Based Air Pollution Monitoring System is used to monitor Air Quality over a web server using the Internet. It will trigger an alarm when the air quality goes down beyond a certain level, which means when there are sufficient amounts of harmful gases in the air like CO2, smoke, alcohol, NH3 and NOx. It will show the air quality in PPM on the show data and the webpage so that they can easily monitor air pollution. The system uses MQ135 and MQ6 sensors for monitoring Air Quality as it detects the most harmful gases and can measure their amount accurately. The system uses an Arduino UNO to connect to an IOT platform and provide information about contaminants to the staff. Based on a microcontroller, the system employs gas sensors, DHT11 temperature and humidity, a display, and a buzzer. Arduino UNO and MQ135 sensors are intended to use as monitoring data and alert systems for air pollution. In the present research, an air pollution prediction model is established with the help of machine learning models. Machine learning techniques such as Q-Learning applied to the dataset. It is simple and fast to alter it to fit our changing needs. Then, the target of this project is to apply the machine learning (Q-Learning) concept for the prediction and analysis of gas sensors' pollution levels so that we can analyze the pollution level due to the pollutant gases based on prediction analysis.**

Keywords: **Machine Learning, Accuracy, Classifier, Arduino UNO, Sensor, Q-Learning.**

INTRODUCTION

Air pollution is one of the most pressing environmental challenges of the 21st century, with far-reaching consequences for human health, ecosystems, and climate change. According to the World Health Organization (WHO), 9 out of 10 people worldwide breathe polluted air, leading to 7 million premature deaths annually due to diseases like asthma, lung cancer, and cardiovascular disorders.

Pollutants such as PM2.5, PM10 (particulate matter), CO_2, NO_2, SO_2, and volatile organic compounds (VOCs) are major contributors to deteriorating air quality. Traditional air quality monitoring systems rely on fixed stations operated by government agencies. While accurate, these systems are expensive, spatially limited, and lack real-time data accessibility. This gap creates a need for low-cost, portable, and IoT-enabled solutions that democratize air quality monitoring and empower individuals and communities to take preventive actions. The integration of Internet of Things (IoT) and Machine Learning (ML) offers transformative potential for environmental monitoring. IoT enables real-time data collection through interconnected sensors, while ML models can analyze historical and real-time data to predict future air quality trends and provide actionable insights. This project bridges the gap between real-time monitoring and predictive analytics, leveraging affordable hardware (Arduino, NodeMCU) and advanced algorithms to create a holistic air quality management system.

LITERATURE SURVEY

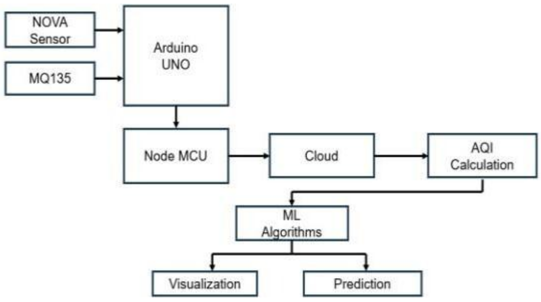
Introduction Air quality monitoring is a crucial aspect of environmental sustainability and public health. With increasing industrialization and urbanization, air pollution levels have risen, necessitating real-time monitoring and predictive analysis. This literature survey explores various methods, technologies, and approaches employed in air quality monitoring. Traditional Air Quality Monitoring Methods Traditional air quality monitoring relies on fixed monitoring stations equipped with high-precision analytical instruments. These stations measure pollutants such as PM2.5, PM10, NO2, SO2, CO, and O3. However, these methods are expensive, require regular maintenance, and lack spatial coverage due to the limited number of stations (Kim et al., 2018). Low-Cost Sensor-Based Monitoring Systems Recent advancements in sensor technology have led to the development of low-cost, portable air quality monitoring devices. Sensors such as MQ135 (for NH3, NOx, alcohol, benzene), Nova PM sensors (for PM2.5 and PM10), and SDS011 (for particulate matter) have gained popularity. These sensors offer cost-effective solutions for localized air quality monitoring, although calibration and accuracy remain challenges (Mead et al., 2013; Castell et al., 2017).

IoT-Based Air Monitoring Systems The integration of the Internet of Things (IoT) in air quality monitoring enables real-time data collection and transmission. Microcontrollers like Arduino and NodeMCU (ESP8266) facilitate data acquisition and communication with cloud platforms such as ThingSpeak. IoT- based systems provide continuous monitoring, remote access, and alerts for pollution levels (Gupta et al., 2020; Kumar et al., 2021).

Machine Learning for Air Quality Prediction Machine learning (ML) techniques are widely used for air quality forecasting. Algorithms such as linear regression, decision trees, support vector machines (SVM), and deep learning models process historical and real-time sensor data to predict pollutant levels. Reinforcement learning methods like Q-learning enhance predictive capabilities by adapting to environmental changes (Zhang et al., 2019; Liu et al., 2021)

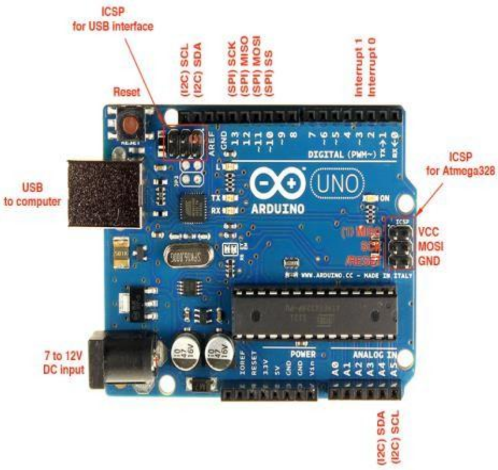
HARDWARE IMPLEMENTATION

Block Diagram:

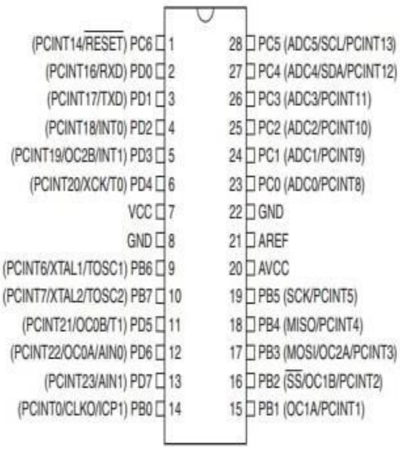


Arduino Uno is a microcontroller board based on the ATmega328 . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC- to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-toserial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. The Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. The board has the following new features: 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.

Arduino pin Diagram:



Pin Configuration:



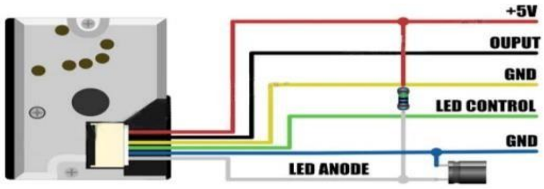
Pin Description:

Pins	Functions
Serial: 0 (RX) and 1 (TX).	Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
External Interrupts: 2 and 3.	Configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
PWM: 3, 5, 6, 9, 10 and 11.	Provide 8-bit PWM output with the analogueWrite() function
SPI:10(SS), 11(MOSI), 12 (MISO), 13 (SCK).	These pins support SPI communication using the SPI library.
LED: 13	There is a built-in LED connected to digital pin13. HIGH pin value - LED on, LOW pin value - LED off.
TWI	A4 or SDA pin and A5 or SCL pin. Support communication using the wire library
AREF	Reference voltage for the analogue inputs. Used with analogueReference().
Reset	To reset the microcontroller

Dust Sensor Pin Diagram:-

A dust sensor is an electronic device designed to detect and measure the concentration of dust particles suspended in the air. These sensors are widely used in air quality monitoring, industrial safety, and environmental applications. By measuring particulate matter (PM) such as PM2.5 and PM10, dust sensors help assess pollution levels and enable preventive actions to maintain clean air.The working principle of a dust sensor depends on its type, but the most common methods include light scattering and infrared absorption. In an optical dust sensor, a light source, usually an the concentration of dust particles suspended in the air. These sensors are widely used in air quality monitoring, industrial safety, and environmental applications. By measuring particulate matter (PM) such as PM2.5 and PM10, dust sensors help assess pollution levels and enable preventive actions to maintain clean air.The working principle of a dust sensor depends on its type, but the most common methods include light scattering and infrared absorption. In an optical dust sensor, a light source, usually an LED or laser, emits light into a sensing chamber. When air containing dust particles passes through this chamber, the particles scatter the light. A photodetector placed inside the sensor detects this scattered light, and the intensity of the scattered light is directly proportional to the number and size of dust particles present in the air. The sensor then converts this optical signal into an electrical signal, which is processed to determine the dust concentration, typically measured in micrograms per cubic meter (µg/m³).Another type of dust sensor operates using the infrared absorption principle. In this method, an infrared (IR) beam is transmitted across a sensing chamber. When dust particles are present, they absorb or block some of the infrared light, reducing the intensity of light reaching the IR detector. By measuring the change in infrared light intensity, the sensor estimates the concentration of dust in the air. This type of sensor is commonly used in industrial environments for monitoring particulate pollution.Dust sensors are crucial in various applications, including air purifiers, HVAC systems, industrial air monitoring, and IoT-based air quality monitoring networks. By providing real-time dust concentration data, these sensors help ensure safe air conditions for both indoor and outdoor environments.

Pin Description:-



Pin name	Description
V_LED	Powers the internal LED (5V)
LED-GND	Ground for LED circuit
LED CONTROL	Controls LED operation (connected-microcontroller)
VCC	Power supply for sensor (5V)
OUTPUT (V0)	Analog output voltage proportional to dust concentration
GND	Ground for sensor circuit

Connection of Dust Sensor with Arduino Uno:-

Dust sensor pin	Arduino Uno pin
Vcc(pin 4)	5V
Gnd(pin 6)	GND
LED control (pin 3)	Digital pin(e.g.,D2)
Output (pin 5)	Analog pin(e.g.,A0)

MQ135 Multi Gas Sensors:

MQ135 Gas Sensor module for Air Quality having Digital as well as Analog output. Sensitive material of MQ135 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exist, The sensors conductivity is more higher along with the gas concentration rising. MQ135 gas sensor has high sensitivity to Ammonia, Sulphide and Benze steam, also sensitive to smoke and other harmful gases. It is with low cost and suitable for different application. Used for family, Surrounding environment noxious gas detection device, Apply to ammonia, aromatics, sulfur, benzene vapor, and other harmful gases/smoke, gas detection, tested concentration range: 10 to 1000 ppm. When air passes over the heated tin dioxide (SnO_2) sensing layer, the sensor reacts to gases by altering its electrical resistance. In clean air, the sensor has high resistance, but when exposed to pollutants, it absorbs the target gases, reducing resistance. The internal circuit converts this resistance change into an analog voltage signal, which is then processed by a microcontroller (e.g., Arduino or Raspberry Pi) to determine air quality levels. The output is often used to calculate the Air Quality Index (AQI), indicating whether the air is safe or polluted.

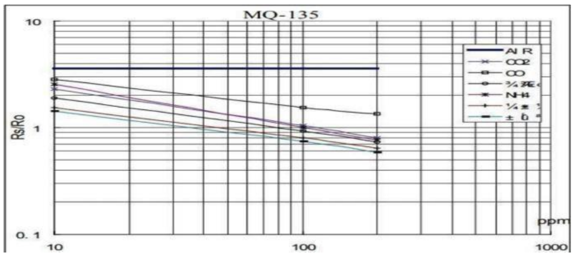
Figure:MQ135 Gas Sensor

Pin Description:

Pin	Pin name	Description
1	Vcc	Used to power the sensor, Generally the operating voltage +5V
2	GND	Used to connect the module to system ground.
3	Digital out	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer.
4	Analog out	This pin outputs 0-5V analog voltage based on the intensity of the gas.

How to use MQ-135 sensor to measure PPM:

MQ-135 gas sensor applies SnO2 which has a higher resistance in the clear air as a gas- sensing material. When there is an increase in polluting gases, the resistance of the gas sensor decreases along with that. To measure PPM using MQ-135 sensor we need to look into the (Rs/Ro) v/s PPM graph taken from the MQ135 datasheet. The above figure shows shows the typical sensitivity characteristics of the MQ-135 for several gases. The value of Ro is the value of resistance in fresh air (or the air with we are comparing) and the value of Rs is the value of resistance in Gas concentration. First you should calibrate the sensor by finding the values of Ro in fresh air and then use that value



Resistance of sensor(Rs): Rs=(Vc/VRL-1)×RL

to find Rs using the below formula: Once we calculate Rs and Ro we can find the ratio and then using the graph shown above we can calculate The equivalent value of PPM for that particular gas.

Wi-Fi module ESP8266:

ESP8266 is Wi-Fi enabled system on chip (SoC) module developed by Espressif system. It is mostly used for the development of the Internet of Things (IoT) embedded applications The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by Shanghai-based Chinese manufacturing company Espressif Systems.

The ESP8266 is capable of either hosting an application or offloading all the WiFi networking functions from another application processor. Each ESP8266 Wi-Fi module comes pre- programmed with an AT command set firmware, now you can simply hook this up to your Arduino device and get as much Wi-Fi ability as a Wi-Fi Shield offers.

ESP8266:
It employs a 32-bit RISC CPU (reduced instruction set computer) based on the Tensilica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz).It has a 64 KB boot ROM, 64 KB instruction RAM, and 96 KB data RAM. External flash memory can be accessed through SPI.To communicate with the ESP8266 module, the microcontroller needs to use AT commands.The microcontroller communicates with the ESP8266-01 module using UART (Universal Asynchronous Receiver/Transmitter) having specified Baud rate.

ARDUINO IDE:
Arduino IDE where IDE stands for Integrated Development Environment – An official software introduced by Arduino.cc, that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go.

Introduction to Arduino IDE:

- Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
- It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
- It is easily available for operating systems like MAC, Windows, and Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.
- A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.
- Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
- The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
- The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.

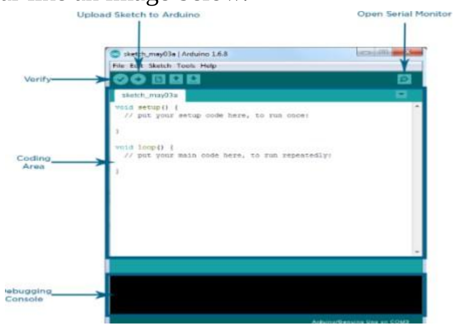
5.1.2How to install Arduino IDE:

You can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you 48 are downloading the correct software version that is easily compatible with your operating system.

If you aim to download Windows app version, make sure you have Windows 8.1 or Windows 10, as app version is not compatible with Windows 7 or older version of this operating system. The IDE environment is mainly distributed into three sections

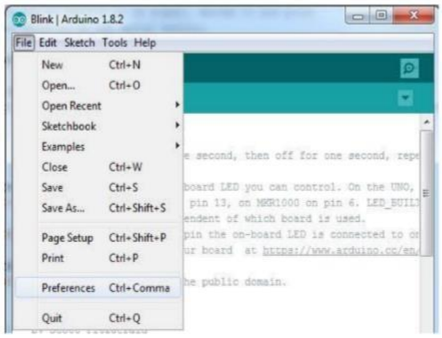
1. Menu Bar
2. Text Editor
3. Output Pane

As you download and open the IDE software, it will appear like an image below.



The bar appearing on the top is called Menu Bar that comes with five different options as follow.

File – You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into.



As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button. And at the end of compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.

- Edit – Used for copying and pasting the code with further modification for font
- Sketch – For compiling and programming
- Tools – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller
- Help – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

The Six Buttons appearing under the Menu tab are connected with the running program as follow. The check mark appearing in the circular button is used to verify the code. Click this once you have written your code. The arrow key will upload and transfer the required code to the Arduino board.

- The dotted paper is used for creating a new file.
- The upward arrow is reserved for opening an existing Arduino project.
- The downward arrow is used to save the current running code.
- The button appearing on the top right corner is a Serial Monitor – A separate pop-up window that acts as an independent terminal and plays a vital role for sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.
- You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor, the output will show as the image below. The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module. More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller, however, there are some dedicated libraries used for calling and executing specific functions on the board.

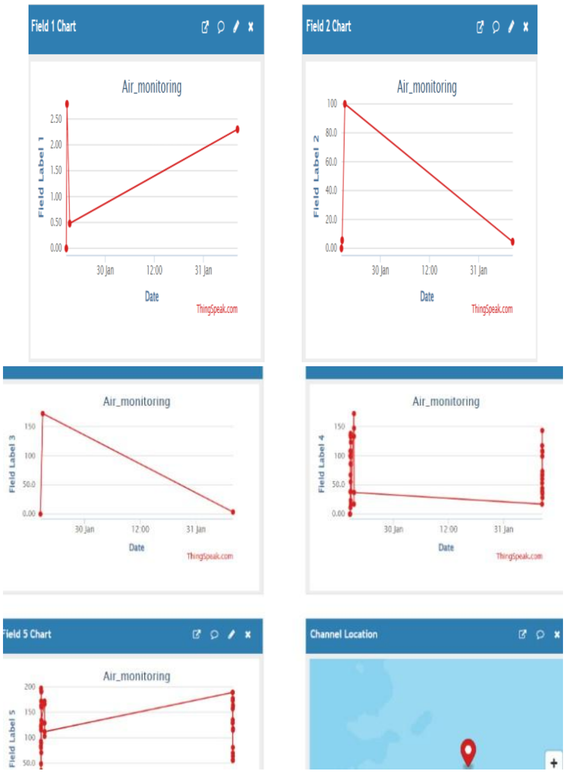
Libraries:

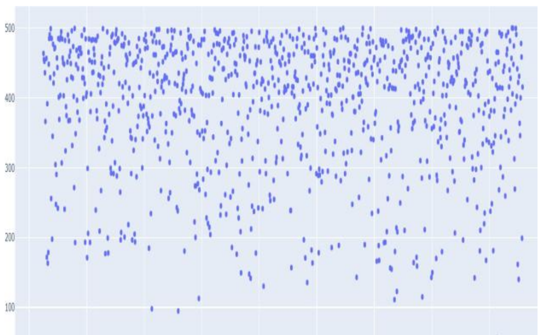
Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library. As you click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, it will appear on the text editor as

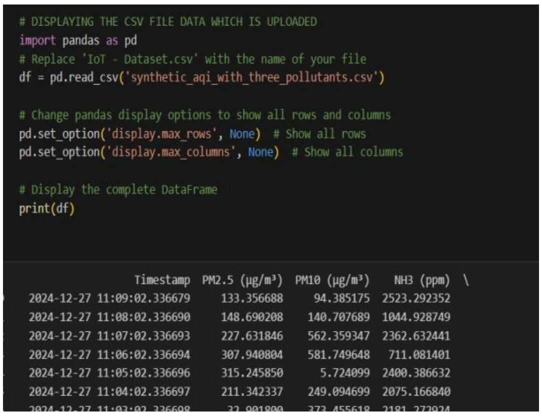
#include <EEPROM>.

Most of the libraries are preinstalled and come with the Arduino software. However, you can also download them from the external sources.

RESULTS ANALYSIS

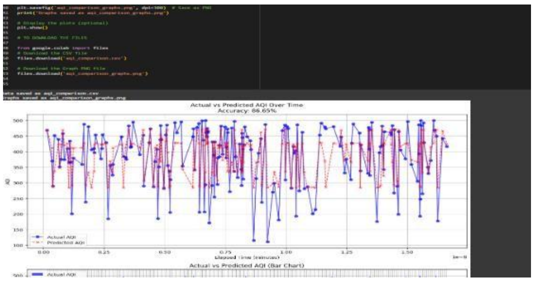


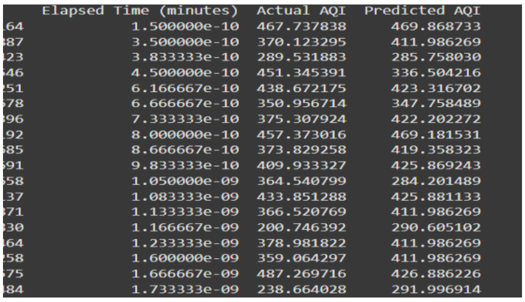




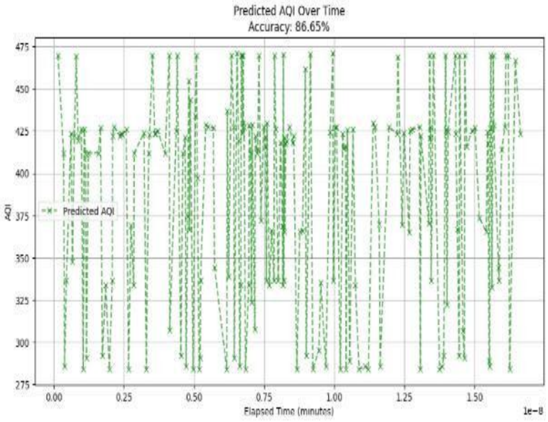
```
# DISPLAYING THE CSV FILE DATA WHICH IS UPLOADED
import pandas as pd
# Replace 'IoT - Dataset.csv' with the name of your file
df = pd.read_csv('synthetic_aqi_with_three_pollutants.csv')

# Change pandas display options to show all rows and columns
pd.set_option('display.max_rows', None)  # Show all rows
pd.set_option('display.max_columns', None)  # Show all columns

# Display the complete DataFrame
print(df)
```

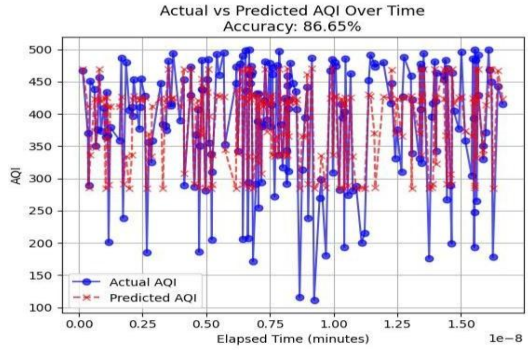
Timestamp	PM2.5 (µg/m³)	PM10 (µg/m³)	NH3 (ppm)
2024-12-27 11:09:02.336679	133.356688	94.385175	2523.292352
2024-12-27 11:08:02.336690	148.690208	140.707689	1044.928749
2024-12-27 11:07:02.336693	227.631846	562.359347	2362.632441
2024-12-27 11:06:02.336694	307.940804	581.749648	711.081401
2024-12-27 11:05:02.336696	315.245850	5.724099	2400.386632
2024-12-27 11:04:02.336697	211.342337	249.094699	2075.166840

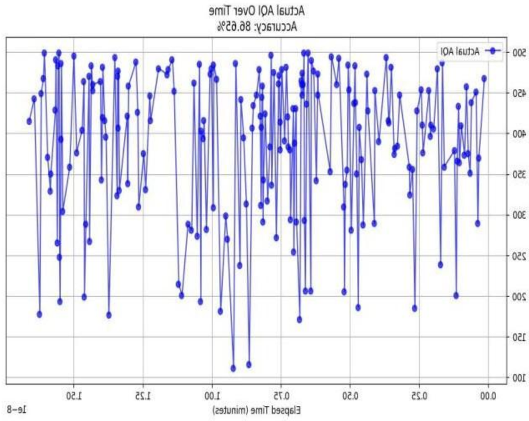


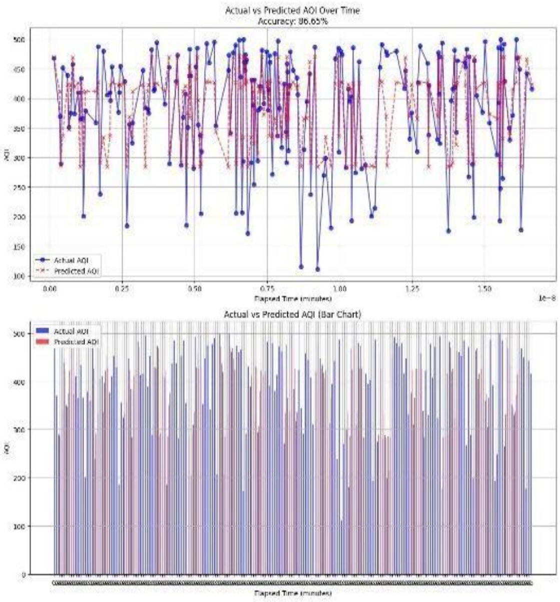


	Elapsed Time (minutes)	Actual AQI	Predicted AQI
64	1.500000e-10	467.737838	469.868733
87	3.500000e-10	370.123295	411.986269
23	3.833333e-10	289.531883	285.758030
46	4.500000e-10	451.345391	336.504216
51	6.166667e-10	438.672175	423.316702
78	6.666667e-10	350.956714	347.758489
96	7.333333e-10	375.307924	422.202272
92	8.000000e-10	457.373016	469.181531
85	8.666667e-10	373.829258	419.358323
91	9.833333e-10	409.933327	425.869243
58	1.050000e-09	364.540799	284.201489
37	1.083333e-09	433.851288	425.881133
71	1.133333e-09	366.520769	411.986269
30	1.166667e-09	200.746392	290.605102
64	1.233333e-09	378.981822	411.986269
58	1.600000e-09	359.064297	411.986269
75	1.666667e-09	487.269716	426.886226
84	1.733333e-09	238.664028	291.996914









APPLICATIONS

- Public Health
- Disease Prevention
- Environmental Protection
- Ecosystem Monitoring
- Urban Planning and Smart Cities
- Traffic Management: Air quality data can inform traffic management strategies, such as rerouting vehicles or promoting public transportation to reduce emissions in congested

areas.

- Green Infrastructure: Cities can use air quality predictions to plan green spaces, such as parks and tree-lined streets, to improve air quality and reduce urban heat islands.
- Industrial Applications
- Compliance Monitoring: Industries can monitor emissions to ensure compliance with environmental regulations and avoid penalties.
- Operational Efficiency: Real-time air quality data can help industries optimize processes to reduce emissions and improve energy efficiency

ADVANTAGES

Real-time Monitoring Early Warning System Low power consumption Improved Public Health Smart City Integration

Data-driven Decision Making Cost-effective & Scalable Community Awarenes

FUTURE SCOPE

Integration with Smart Cities:-The system can be integrated with smart city infrastructure to provide real-time air quality data to municipal authorities. Automated traffic and industrial control measures based on pollution levels.

Drone-Based Air Quality monitoring:-Deploying drones equipped with air quality sensors to monitor pollution over larger areas, including industrial zones and forests. Real-time mapping of pollution dispersion. Integration with Healthcare Systems:-

Integration with Healthcare Systems:- Connecting air quality data with hospital systems to provide real-time alerts for asthma and respiratory patients. Personalized health recommendations based on pollution levels. Enhanced Sensor Accuracy and Coverage:- Using multi-sensor fusion techniques to improve the accuracy of air quality measurements. Expanding sensor network coverage to remote and rural areas.

Public Awareness and Mobile App Integration:- Creating mobile applications with real-time air quality updates and health recommendations. AI-based notifications for users in highly polluted areas.

REFERENCES

[1] Liu, Y., & Uthra, R. A. (2020). An IoT-Based Real-Time Air Quality Monitoring System for Electric Bike Warehouse. International Journal of Advanced Science and Technology. Available at: sersc.org

[2] Barthwal, A., & Acharya, D. (2021). An IoT-Based Sensing System for Modeling and Forecasting Urban Air Quality. Wireless Personal Communications. Available at: Springer

[3] Hawari, H. F., Zainal, A. A., & Ahmad, M. R. (2019). Development of Real-Time Internet of Things (IoT) Based Air Quality Monitoring System. Indonesian Journal of Electrical Engineering and Computer Science. Available at: IAES Journal

[4] Benammar, M., Abdaoui, A., Ahmad, S. H. M., Touati, F., & Kadri, A. (2018). A Modular IoT Platform for Real-Time Indoor Air Quality Monitoring. Sensors, 18(2), 581. Available at: MDPI

[5] Jo, J. (2020). Development of an IoT-Based Indoor Air Quality Monitoring Platform. Journal of Sensors. Available at: Wiley Online Library

[6] Pramanik, J., Samal, A. K., Pani, S. K., & Chakraborty, C. (2022). Elementary Framework for an IoT-Based Diverse Ambient Air Quality Monitoring System. Multimedia Tools and Applications. Available at: Springer

[7] Buelvas, J., Múnera, D., Tobón, D. P. V., Aguirre, J., & Gaviria, N. (2023). Data Quality in IoT-Based Air Quality Monitoring Systems: A Systematic Mapping Study. Water, Air, & Soil Pollution. Available at: Springer

[8] Karnati, H. (2023). IoT-Based Air Quality Monitoring System with Machine Learning for Accurate and Real-Time Data Analysis. arXiv preprint arXiv:2307.00580.

[9] Zhang, X., Li, Y., & Wang, L. (2021). Hybrid Models for Air Quality Prediction Combining IoT and Machine Learning Techniques. Applied Soft Computing, 104, 107222.

[10] Jain, S., & Bhatnagar, P. (2022). Air Quality Monitoring Using Neural Networks: A Machine Learning Perspective. Sensors and Actuators B: Chemical, 349, 130763.

[11] Alshamsi, A., & Arshad, K. (2023). An Intelligent Edge- Deployable Indoor Air Quality Monitoring and Activity Recognition Approach.

[12] Singh, P., & Verma, K. (2024). Applications of Machine Learning and IoT for Outdoor Air Pollution Monitoring and Prediction: A Systematic Literature Review.