AIR POLLUTION MONITORING AND FORECAST OF POLLUTION LEVEL USING IOT AND ML

[1]Mohammed Jayeed, [2]Mohammed Shahbaz , [3]Shaik Tabrez Ahmed
[1]Assistant professor, [2] [3]Students
Department of Electronic and communication Engineering
ISL engineering college, bandlaguda, Telangana
Email: mdshahbaz@gmail.com, tabraizahmed128@gmail.com

Abstract:- Air pollution poses a significant threat to public health, the environment and in recent times is increasing at an alarming rate, necessitating the development of efficient monitoring and prediction systems. This paper presents an innovative approach leveraging the Internet of Things (IoT) and Machine Learning (ML) techniques for real-time air pollution monitoring and prediction. The proposed system integrates a network of IoT sensors strategically placed in urban areas to continuously measure key air quality parameters such as particulate matter (PM), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), ozone (O3), Temperature and Humidity. A microcontroller processes the data from the properly calibrated sensors and sends them to realtime cloud storage through a Wi-Fi module. A remote server then fetches and analyzes the data. We then determine the prime pollutant from the data, which is Particulate Matter. Next, we use the Autoregressive Integrated Moving Average (ARIMA) model to predict the pollution levels from harmful gasses & the Air Quality Index (AQI) of the next day with high accuracy. Precisely, we predict the 24-hourly observations of the following day after training our optimized model with 144 hourly observations of the previous six days. We then evaluate the model with a MAPE (Mean Absolute Percentage Error) score, which ensures that our model is working properly. This system offers proactive decision-making capabilities, enabling stakeholders to issue alerts, implement control measures, and optimize urban infrastructure to mitigate pollution. Visualization tools and reporting mechanisms provide intuitive insights for policymakers and the public.Through continuous evaluation and refinement, the proposed system aims to enhance air quality management, contributing to healthier and more sustainable communities. Therefore we believe that our solution will be helpful if implemented at large scale.

Keywords: Air Pollution, Monitoring, Forecasting, Internet of Things (IoT), Big Data, Machine Learning (ML), ARIMA, Wireless Sensor Network (WSN)

1. INTRODUCTION:- Air quality refers to the degree to which the air is free of pollutants in a particular location at a given time. Natural pollution sources include wildfires, volcanoes and dust storms while man-made sources include emissions from vehicular exhaust, power plants factories etc. The concerns over air quality is increasing daily due to the pollution of air caused mainly due to human action which leads to major health concerns. It is attributed as the main environmental health issue and as a major cause of premature human death causing heart disease, stroke, lung disease and cancer.

Current air quality monitoring methods include continuous monitoring methods, gravimetric particulate methods and passive monitoring methods. Air pollutants require continuous monitoring due to its high resolution to obtain hourly or daily average concentrations. Air quality prediction using conventional mathematical and statistical approaches in the past have been inefficient until the advancements in the field of big data and machine learning. With the development in this field more research adopted this methodology to efficiently forecast air quality. Current data driven methods of forecasting readings from an air quality monitoring station include the use of a neural network for global factors and the use of linear regression based temporal predictor for local factors.

In order to bring the level of pollution to a minimum, there is a necessity of cost-effective, accurate mechanisms to monitor air quality in real time and to forecast air quality parameters to conduct studies on ambient air quality.

For this reason, in this project, we would like to build an IoT device that can monitor and predict the air quality around us. We will be using four different sensors to track the air quality around us. PM2.5 (Particulate Matter), CO (Carbon monoxide), NH3 (Ammonia), and Temperature & Humidity will all be measured. Next, the data will be processed by a microcontroller and sent to a server with the help of a WiFi module. Finally, we will show the user the concentrations of the gasses in ppm (part per million), overall air quality by using AQI, and future air quality by using machine learning models. The main contributions of this paper can be summarized as follows:

- We propose a two-in-one device combining IoT and Machine Learning that can accurately monitor pollution levels as well as forecast future pollution levels.
- Our solution focuses on monitoring and prediction.
- The device is capable of tracking the surrounding AQI with high accuracy using four different important parameters.
- The model can predict future pollutant levels with an accuracy of 90% or more.

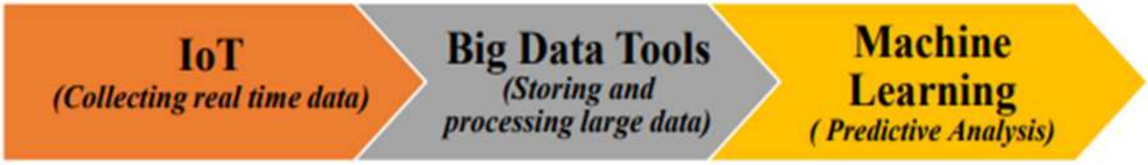

IoT
(Collecting real time data)
Big Data Tools
(Storing and processing large data)
Machine Learning
(Predictive Analysis)

Fig-1: Technologies used in monitoring and prediction models.

2. LITERATURE REVIEW:-

2..1 IoT for monitoring air quality:-

IoT (Internet of Things) can be defined as “An open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment”. IoT is a popular concept in providing successful solutions to air quality monitoring. The “GasMobile” system, which is a participatory air quality monitoring system, has been implemented using a self-developed air quality sensing device in combination with android smartphones. The concept is that the citizen participates in the collection of air quality information. Devarakonda proposed an air quality system consisting of sensing devices deployed on vehicles such as public transportation and personal vehicles which consists of an arduino microcontroller, PM sensor, CO sensor and a modem which will upload data to the server. Yang & Li developed an air quality system with a microprocessor, CO sensor and VOC sensor which takes readings when the user executes a mobile application which is then transmitted to the mobile phone. However, the system measures air quality parameters on demand only.

2.2 Air quality prediction:-

Fluctuation of air quality parameters is a complex phenomenon since it is a result of a combination of emissions, meteorological, demographic and terrain factors. Pollutants due to vehicular emissions such as particulate matter (PM10, PM2.5) and Carbon Monoxide (CO) show temporal variations throughout the day with time and the peak in vehicular activity.

An air quality forecasting model gives a deterministic description of the greater air problem and a tool used to explain the relationship between air quality determining parameters.

Air quality prediction using a hidden semi-Markov model for the prediction of PM2.5 proved to provide reasonable accuracy while the Bayesian approach to air quality prediction was better than the semi-Markov approach.

Overfitting is a challenge in machine learning models where the model gives highly accurate results during training but poor results during testing. Non-linear models are however superior to linear models because they capture non-linear relationships in the data set of pollutant concentrations.

3. METHODOLOGY:-

The high-level overview of our project has been depicted using a workflow diagram in Fig. 2. We have connected four sensors (two gas, one dust, and one temperature & humidity) and a Wi-Fi module with the microcontroller and placed it inside a house. Data from the sensors is then sent to the real time cloud storage (ThingSpeak) through the Wi-Fi module. It is then fetched for data analysis and data modeling to forecast future values. A hardware circuit diagram setup of our project has been depicted in Fig. 3.

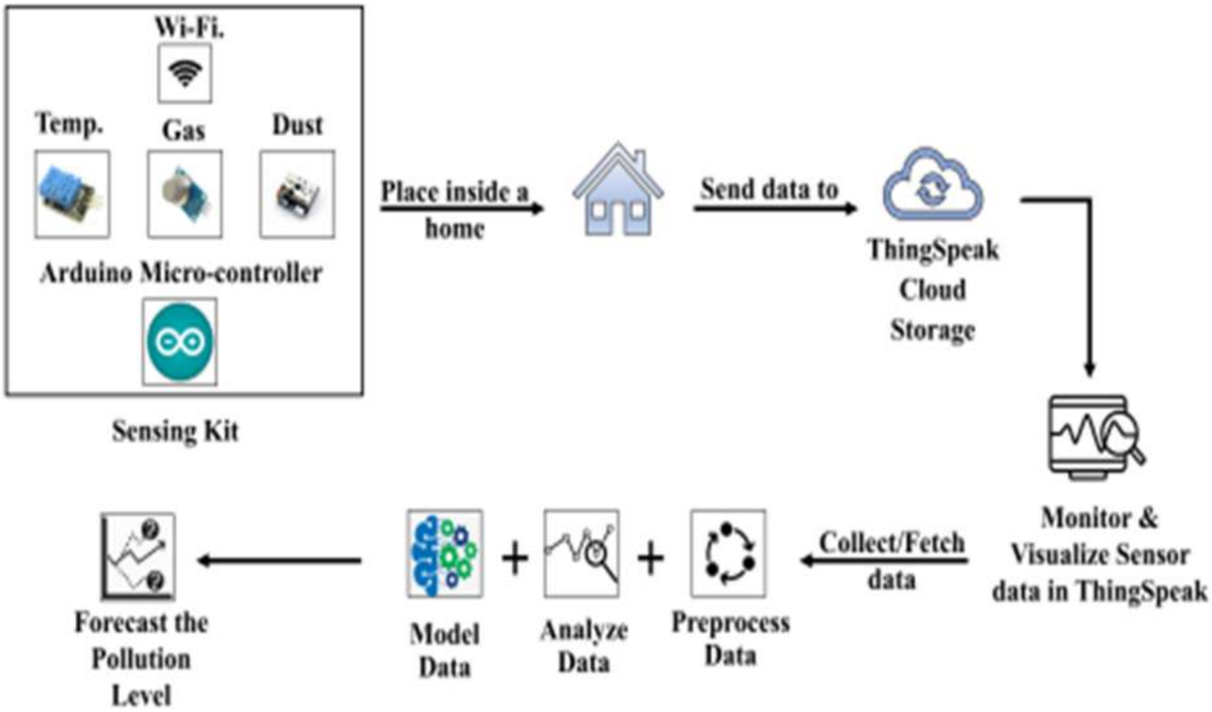


Fig-2: Workflow Diagram

3.1 **SYSTEM ARCHITECTURE:-**

We have selected Arduino Uno with a processor as our microcontroller. Fig. 3 shows the system diagram of our project. We have connected four sensors with the microcontroller. MQ-135 to measure NH3, MQ-7 to measure CO, Sharp Dust Sensor to measure PM2.5, and DHT11 to measure Temperature & Humidity. These sensors send digital signals 0−1023 to the Arduino board by converting the analog voltage (0V − 5V) variations. Arduino has its IDE and its programming language, which is similar to the programming language C/C++. In the Arduino IDE, we wrote the necessary codes and uploaded them into our Arduino board to fetch the data from the sensors. Next, after setting up the Wi-Fi module, our data goes to real-time Cloud Storage (ThingSpeak). From there, using the read API, data is fetched for preprocessing. After that, various models were evaluated, and the best model that suits our dataset, i.e., ARIMA (Autoregressive Integrated Moving Average), was selected. Then we train our model to predict the future pollutant levels of all the parameters (CO, NH3, temperature, humidity & PM2.5). Finally, various experiments were held to check the integrity of our sensor data and the accuracy of our prediction model, which are explained in the experiment and results section.

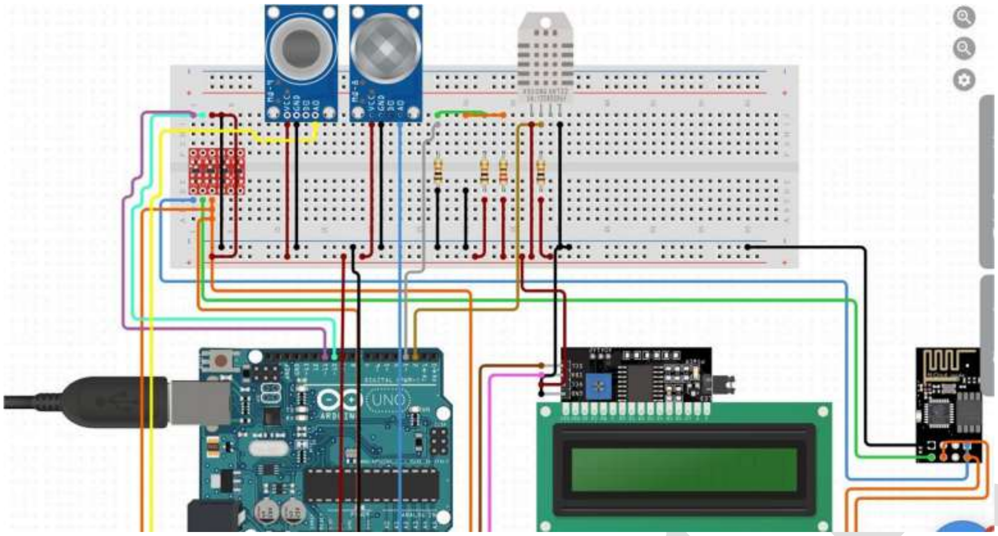


Fig-3: Hardware Setup

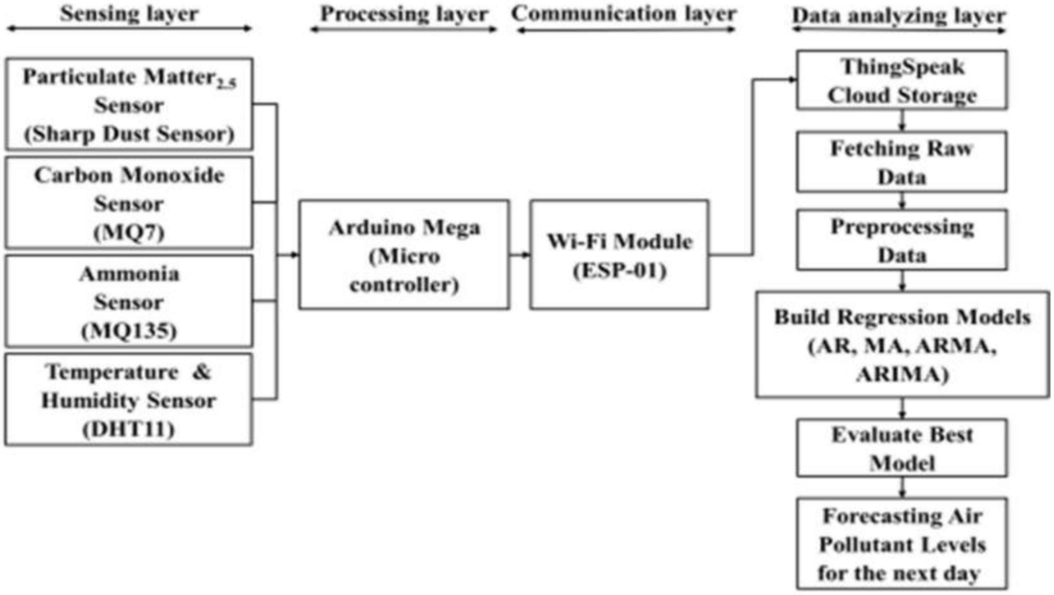

Sensing layer
Processing layer
Communication layer
Data analyzing layer
Particulate Matter$_{2.5}$ Sensor (Sharp Dust Sensor)
Carbon Monoxide Sensor (MQ7)
Ammonia Sensor (MQ135)
Temperature & Humidity Sensor (DHT11)
Arduino Mega (Micro controller)
Wi-Fi Module (ESP-01)
ThingSpeak Cloud Storage
Fetching Raw Data
Preprocessing Data
Build Regression Models (AR, MA, ARMA, ARIMA)
Evaluate Best Model
Forecasting Air Pollutant Levels for the next day

Fig-4:Software Diagram

3.2 HARDWARE AND SOFTWARE REQUIREMENTS:-

(1) Arduino Mega 2560 Microcontroller, (2) Temperature & Humidity Sensor (DHT11), (3) CO Sensor (MQ-7), (4) NH3 Gas Sensor (MQ-135), (5) Optical PM2.5 Dust Sensor (Sharp

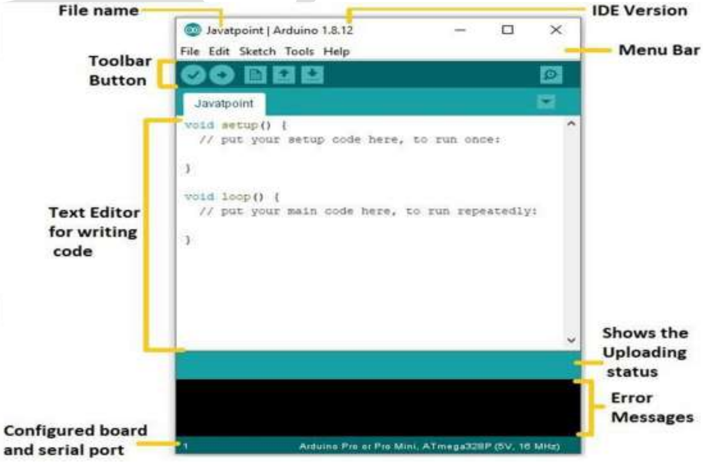
GP2Y1010AU0F), (6) Wi-Fi Module (ESP-01), (7) Breadboard Jumpers wires, (8) Capacitor & Resistor, (9) Arduino IDE, (10) ThingSpeak Real-Time Database, (11) Python Programming Language

3.2.1 NodeMCU V3

NodeMCU V3 is an open-source ESP8266 development kit, armed with the CH340G USB- TTL Serial chip. It has firmware that runs on ESP8266 Wi-Fi SoC from Espressif Systems. Whilst cheaper, CH340 is super reliable even in industrial applications. It is tested to be stable on all supported platforms as well. It can be simply coded in Arduino IDE. It has a very low current consumption between 15 µA to 400 mA.

3.2.2 Arduino IDE

The Arduino IDE is open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment. The program or code written in the Arduino IDE is often called sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

File name
IDE Version
Javatpoint | Arduino 1.8.12
File Edit Sketch Tools Help
Menu Bar
Toolbar Button
Javatpoint
Text Editor for writing code
Shows the Uploading status
Error Messages
Configured board and serial port
1
Arduino Pro or Pro Mini, ATmega328P (5V, 16 MHz)

Fig 5 Arduino IDE

3.2.3 DHT11 with Arduino IDE: DHT11 sensor comes precalibrated from the factory. It has a built-in library called "DHT sensor library," which is used to convert the sensor values from analog to digital so that we can read the values of temperature and humidity.

3.2.4 MQ-135 & MQ-7 with Arduino IDE: MQ-135 and MQ7 gas sensors are used to measure NH3 and CO respectively. These two gas sensors work on the same principle.

They have a built-in resistor (Rs) whose resistance varies based on the gas concentration. If the gas concentration is high, resistance decreases, and if the concentration is low, then resistance increases. To calculate gas concentration from resistance variation, there is a log-to-log plot in the sensor datasheet. The plot has the Rs/Ro ratio on the y-axis and the gas concentration in ppm on the x-axis. Here Ro is the sensor's resistance in the fresh air, whose value needs to be calibrated depending on the area it is kept in. On the other hand, Rs is the variable resistance that determines the concentration of gas. After calibrating Ro and getting Rs from the sensors, Rs/Ro is calculated. Next, this ratio, along with the slope and intercept of the log-to-log plot, is used to calculate the concentration of gas in ppm, as shown in (1). This ppm value is then converted to µg/m3 . Moreover, another resistor called load resistor (RL) is used to adjust the sensor's sensitivity to changes in gas concentration. $\text{ppm} = (\log 10(\text{Rs/Ro}) - \text{y intercept})/\text{slope}$ (1)

3.2.5 Sharp Dust Sensor with Arduino IDE: The Sharp dust is used to measure PM2.5. It comes pre-calibrated from the factory. To get the dust density, analog values from the sensor are first converted to voltages. Next, this sensor voltage, along with the slope and y-intercept of the graph in the sensor datasheet, is used to get the dust density in µg/m3 , as shown in (2).

$$\text{dust density} = ((\text{slope} \times \text{sensor voltage}) - \text{y intercept}) \times 1000$$

3.3 Step to selecting Model for Prediction:-

Step-1: Dataset : We have manually curated our custom dataset with the help of the values from the sensors. Since we are performing univariate time-series forecasting, each parameter will have its own time-series dataset. So, for our five parameters (CO, NH3, PM2.5, Temperature & Humidity), we have prepared five time-series datasets. Besides, all of our time series datasets consist of hourly observations. So, for each of the five parameters, we will be forecasting the next 24 hourly observations by leveraging the hourly observations of the past six days.

Step-2: Data processing: Data preprocessing may be defined as converting the raw data from sensors into a well-organized functional format by trimming out garbage values, noises and arranging them according to the needs of a model. In our case, while collecting raw data from our sensors, we had to remove null and missing values. Besides, we also make the data points univariate, i.e., a function of only one variable. Each parameter has an interval of 3.5 minutes between two data points. So, in 1 hour, every parameter has 17 data points which are then averaged to deduce the hourly values of the parameters.

Step-3: Model evaluation: There are various models for time-series data prediction. But before selecting a model, it is mandatory to see if the model suits our dataset or not. For this reason, we have omitted some models as they would not give us accurate scores. Auto Regression and ARMA (Autoregressive Moving Average) were also not suitable since these models require a stationary dataset, and our dataset has some non-stationary parameters. Besides, we tried using the models Linear/Polynomial Regression and LSTM (Long Short-Term Memory) but dropped them as they gave inaccurate results. The main reason was the lack of observations in both cases. Lastly, we also didn’t use Seasonal ARIMA (Auto-Regressive Integrated Moving Average) and ARIMAX because of insufficient data and our data not being multivariate

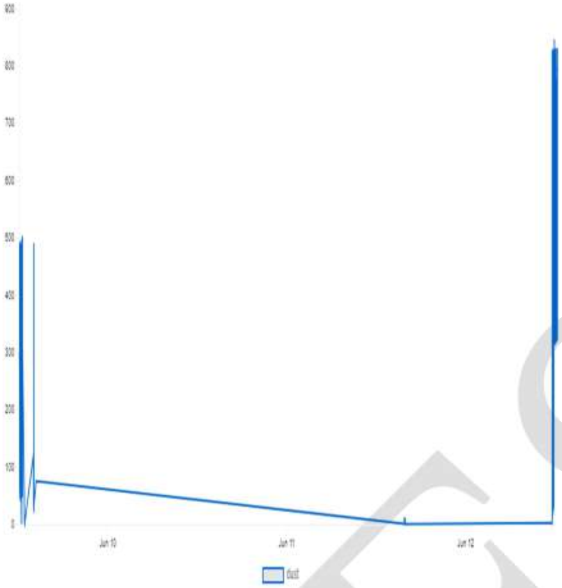
Step-4: Train Module: Train them by taking the collected data for seven days to predict pollution level.

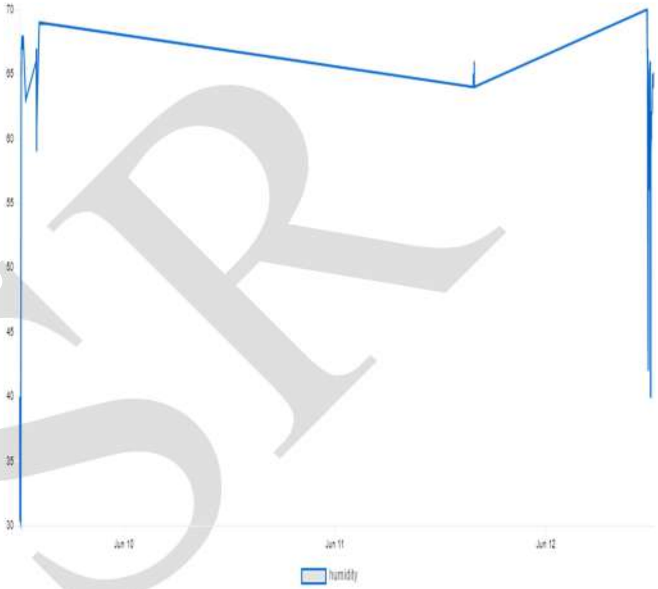
4. EXPERIMENT & RESULT:

We perform various experiments with our dataset to draw certain conclusions. At first, we experimented to observe the parameter & AQI values for seven consecutive days and determine the prime pollutant. Next, we compare our indoor data with that of an outdoor weather station and analyze the differences. Finally, we test the accuracy of our prediction model by training it with a dataset of a specific period.

7-day values of all the parameters. All of these were calculated after properly calibrating the sensors. We fail to observe any proper trend in the data for all the parameters. Moreover, we see that overall AQI was determined by PM2.5 for the majority of the cases, although NH3 comes at a close second. The AQI values reflect that the air quality was “poor” or “very poor” for all the days. This is caused by several reasons. Typically, indoor PM2.5 levels are estimated to be the same as outdoor PM2.5 levels, provided that the house is free from any strong combustive activities like smoking. So, in our case, the high indoor PM2.5 levels are mainly due to pollution from the outdoors.

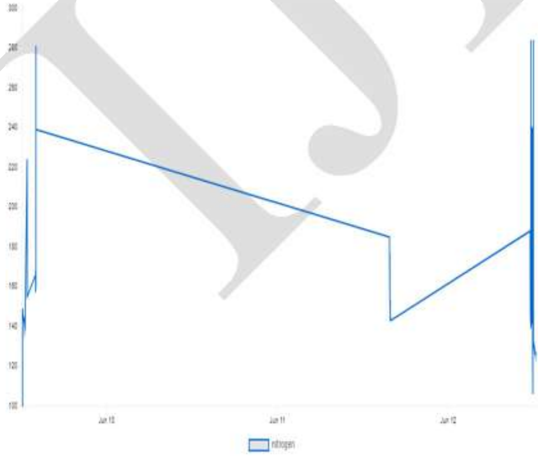
4,1 Sensor value graphs:-

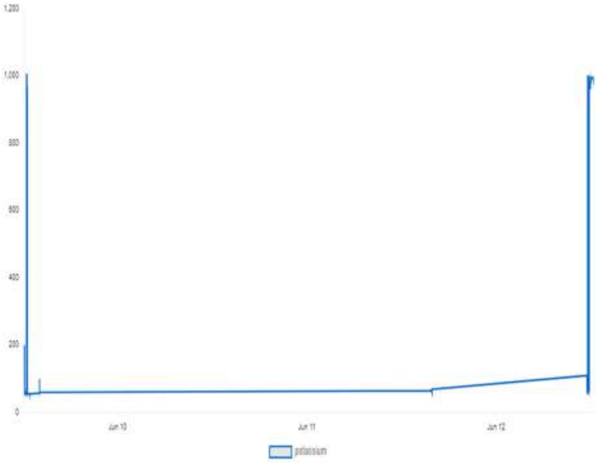




Fiq : 1 Dust sensor graph

Fiq-2 Humidity sensor

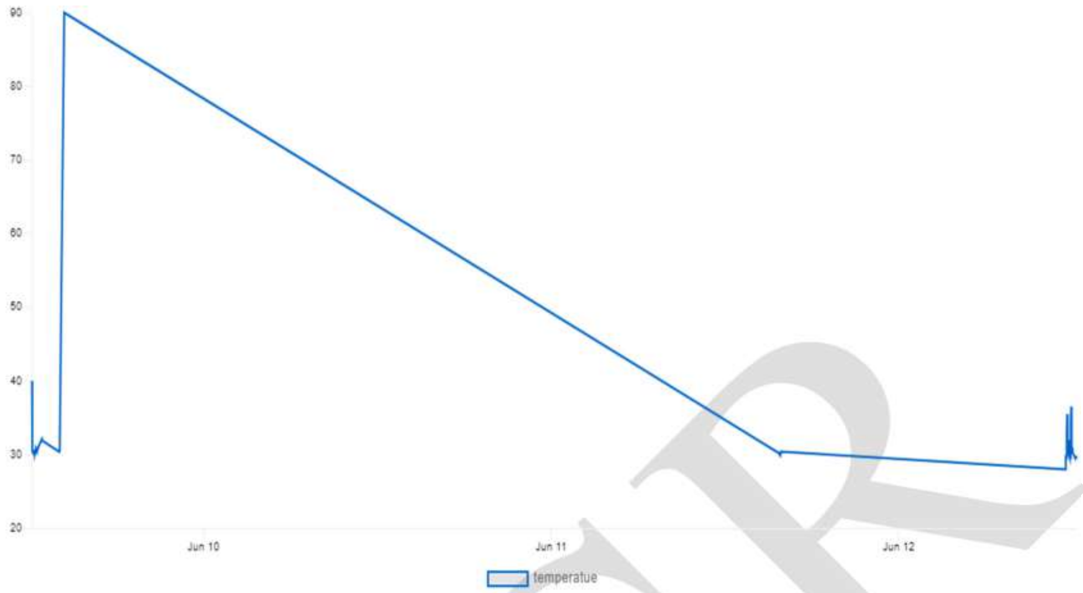




graph

Fiq-3 Carbon monoxide sensor graph

Fiq-4 Ammonia sensor graph



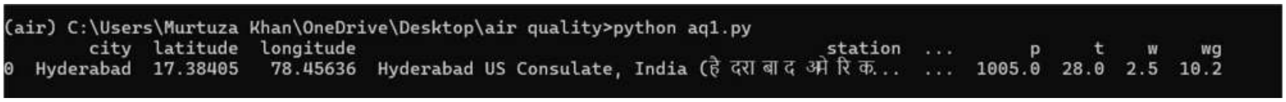
Fiq-5 Temperature sensor graph

Table column for sensor values

S.NO	Temperature(degree centigrade)	Humidity (Relative percentage)	CO (ppm)	NH3 (ppm)	Dust (µg/m3)
1.	33	70 %	129	990	650
2.	31	76 %	126	1000	745
3.	32	73 %	145	998	546
4.	35	65 %	150	991	435
5.	40	60 %	134	982	754

4.2 Prediction result:-

The below shows the predicted population value through weather and dataset



```
(air) C:\Users\Murtuza Khan\OneDrive\Desktop\air quality>python aq1.py
       city  latitude  longitude                                     station  ...       p     t    w    wg
0  Hyderabad  17.38405   78.45636  Hyderabad US Consulate, India (हे दरा बा द अमे रि क...  ...  1005.0  28.0  2.5  10.2
```

5. CONCLUSION:-

We have accomplished the task of calculating various pollutant parameters (CO, PM2.5, NH3, temperature & humidity) accurately after properly calibrating the sensors. Besides, we have also built a system through which anyone can monitor pollution from anywhere in the world by sending data to a cloud server via a Wi-Fi module. Finally, we have successfully implemented a model to train our dataset and have achieved an accuracy of 90% or more in forecasting all the air pollutants levels. Controlling air pollution is a key factor in achieving a sustainable and healthy environment. For this reason, measures should be undertaken to raise awareness across all levels. We believe the solution we have built serves this purpose. In the future, we hope to commercialize our project by leveraging a better hyperparameter tuned model and improved sensors.

6. REFERENCES:-

[1] "Air pollution." [Online]. Available: http://surl.li/opda

[2] U. Gehring, A. H. Wijga, M. Brauer, P. Fischer, J. C. de Jongste, M. Kerkhof, M. Oldenwening, H. A. Smit, and B. Brunekreef, "Traffic Related air pollution and the development of asthma and allergies during the first 8 years of life," American journal of respiratory and critical care medicine, vol. 181, no. 6, pp. 596–603, 2010.

[3] Z. J. Andersen, M. Hvidberg, S. S. Jensen, M. Ketzel, S. Loft, M. Sørensen, A. Tjønneland, K. Overvad, and O. Raaschou-Nielsen, "Chronic obstructive pulmonary disease and long-term exposure to traffic-related air pollution: a cohort study," American journal of respiratory and critical care medicine, vol. 183, no. 4, pp. 455–461, 2011.

[4] R. D. Brook, S. Rajagopalan, C. A. Pope III, J. R. Brook, A. Bhatnagar, A. V. Diez-Roux, F. Holguin, Y. Hong, R. V. Luepker, M. A. Mittleman et al., "Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the american heart association," Circulation, vol. 121, no. 21, pp. 2331–2378, 2010.

[5] M. Goldberg, "A systematic review of the relation between long-term exposure to ambient air pollution and chronic diseases," Reviews on environmental health, vol. 23, no. 4, pp. 243–298, 2008.

[6] Z. J. Andersen, L. C. Kristiansen, K. K. Andersen, T. S. Olsen, M. Hvidberg, S. S. Jensen, M. Ketzel, S. Loft, M. Sørensen, A. Tjønneland et al., "Stroke and long-term exposure to outdoor air pollution from nitrogen dioxide: a cohort study," Stroke, vol. 43, no. 2, pp. 320–325, 2012.

[7] https://gaslab.com/blogs/articles/carbon-monoxide-levels

[8] https://www.instructables.com/Measuring-Humidity-Using-Sensor-DHT11

[9] https://pdf1.alldatasheet.com/datasheet-pdf/view/1307647/WINSEN/MQ135.html

[10] https://components101.com/development-boards/nodemcu-esp8266-pinout-features- and-datasheet