A Project Report on

"An IoT based Sensing System for Monitoring and Forecasting Urban Air Quality"

Submitted to

Shri Guru Gobind Singhji Institute of Engineering and Technology, Vishnupuri, Nanded.
(An Autonomous Institute of Govt. of Maharashtra.)

EDUCATION OF HUMAN POWER FOR TECHNOLOGICAL EXCELLENCE

For the partial fulfilment for the award of the degree of

Bachelor of Technology in Electronics and telecommunication Engineering

Submitted by

Ms. Gangasagar Pikle	**(2016BEC049)**
Ms. Gangubai Malegaonkar	**(2019BEC001)**
Ms. Rashi Reddy	**(2019BEC017)**
Ms. Sneha Chakurkar	**(2019BEC061)**

Under the Guidance of

Dr. DHARMPAL DOYE

(Professor)

Department of Electronics and telecommunication Engineering

Shri Guru Gobind Singhji Institute of Engineering and Technology, Vishnupuri, Nanded

(2022- 2023)

"An IoT based Sensing System for Monitoring and Forecasting Urban Air Quality"

Submitted to

Shri Guru Gobind Singhji Institute of Engineering and Technology, Vishnupuri, Nanded.
(An Autonomous Institute of Govt. of Maharashtra.)

EDUCATION OF HUMAN POWER FOR TECHNOLOGICAL EXCELLENCE

For the partial fulfilment for the award of the degree of

Bachelor of Technology in Electronics and telecommunication Engineering

Submitted by

Ms. Gangasagar Pikle	**(2016BEC049)**
Ms. Gangubai Malegaonkar	**(2019BEC001)**
Ms. Rashi Reddy	**(2019BEC017)**
Ms. Sneha Chakurkar	**(2019BEC061)**

Dr. Dharmpal Doye (Professor)	Mr. Sanjeev Bonde	Dr. Milind Bhalerao (Asst. Professor)
Project Guide	**Project Coordinator**	**Head of Dept.**

Shri Guru Gobind Singhji Institute of Engineering and Technology, Vishnupuri, Nanded.

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Certificate

This is to certify that the project titled "An IoT based Sensing System for Monitoring and Forecasting Urban Air Quality." Is the bonafide work carried out by Gangasagar Pikle (), Gangubai Malegaonkar (), Rashi Reddy () and Sneha Chakurkar () under the guidance of our guide prof. Dr. Dharmpal Doye during the academic year 2022-23, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Electronics and telecommunication Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Mr. Milind Bhalerao
(Head Dept. of EXTC)

Prof. Dr. Dharmpal Doye
(Project Head)

Mr. Sanjiv Bonde
(Project Coordinator)

Dr. Mahesh Kokare
(Director)

External Examiner(S):

Place: Nanded
Date: /05/2023

DECLARATION

I hereby declare that the project entitled "Air Quality Prediction System" submitted for the B.Tech. (EXTC) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Gangasagar Pikle (2016BEC049)

Gangubai Malegaonkar (2019BEC001)

Rashi Reddy (2019BEC017)

Sneha Chakurkar (2019BEC061)

Place: Nanded

Date:

ACKNOWLEDGEMENT

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my respected guide Prof. Dharmpal Doye, EXTC Department, SGGSIE&T, Nanded, for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behaviour are sincerely acknowledged. Our heartfelt thanks to sir for the unlimited support and patience he has shown towards us, without which this work would not have finished in a proper and timely manner.

We are also thankful to all the faculty members in the Dept. of Electronics and Telecommunication Engineering, SGGSIE&T, Nanded, for sharing their knowledge and expertise that helped us throughout our project work. We would also like to thank all of our fellow batchmates, for their help in all matters, for the sleepless nights of working together before deadlines, and for all the fun moments we shared.

Table of Contents

8.Appendices

9.References

ABSTRACT

India has been ranked 8th most polluted country in the world. The air quality of certain cities in the has been reported to be far above the safe limits. Our project aims to bring forth insights into the aspects that affect the quality of the air. It includes the deployment of IoT devices such as sensors and actuators in urban areas to collect data on air quality parameters. These parameters include carbon monoxide (CO), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), relative humidity (RH), and temperature (Temp).

The process of Sensor Data Collection this involves gathering data from the deployed sensors, which measure the aforementioned air quality parameters. The collected data needs to be transmitted and communicated efficiently, leading to the abstraction of Data Transmission and Communication. This aspect encompasses the communication protocols and mechanisms utilized to transmit the sensor data from the IoT devices to a central data processing unit. Once the data is received, the abstraction of Data Processing and Analysis comes into play. This includes cleaning and aggregating the data, as well as performing analysis to derive meaningful insights regarding urban air quality patterns and trends. Real-time Air Quality Monitoring is another important abstraction, involving the continuous monitoring of urban air quality using the collected sensor data. This may involve calculating air quality indices, such as the Air Quality Index (AQI), to provide a comprehensive measure of air quality in specific locations.

The project also incorporates Forecasting and Predictive Analytics. By utilizing historical data and advanced analytical techniques like machine learning algorithms, it becomes possible to forecast future air quality conditions. The aim is to provide early warnings and predictions regarding potential air quality issues in urban areas.

These abstractions form the core components and functionalities of the project, providing an overview of the key aspects involved in the monitoring and forecasting of urban air quality using an IoT-based sensing system.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background

Air pollution has become a serious environmental issue. It is responsible for hundreds of fatalities each year and it poses a serious threat to human health and environment. It leads to global warming, greenhouse effect and it also causes respiratory problems like asthma, lung cancer etc. It is important to predict the quality of the air to regulate air pollution. Air quality index (AQI) is a measure of air quality which describes the level of air pollution. \

To deepen our understanding of air pollution so as to devise effective measures to curb the hazards caused by it, we need to study the air quality data. A number of insights can be drawn from the data like, what are the environmental factors affecting air quality, how air quality changes over the day, is there a time of the day when air quality is critically bad. Same is the case with PM 10 .

The project deals with deploying numerous sensors that return data on regular intervals. These sensors are not required to be state of the art, and their readings can have poor accuracy and precision. Further, these readings can deteriorate over time. Nearly all air quality sensors deployed over the world face these same problems. The project deals with taking these readings, and using an extremely accurate sensor to correct them.

As a lot of the current air quality sensors deployed over the world are highly inaccurate, it would be more effective to develop algorithms to correct their readings than to replace them with new sensors (which will also lose their precision and accuracy over time). Machine learning algorithms can be used for this.

1.2Air Quality

Worldwide, many cities continuously assess the air quality using various monitoring techniques to record the concentrations of the pollutants in the air. Air quality can be defined as the measurement of quality of the air we breathe and the concentrations of the pollutants in the air that can cause various health issues. Air quality can be used for various purposes such as the communication of air quality with the public, to plan strategies that can be used to reduce air pollution, and to monitor short term and long term trends. [2]

Air quality can be measured using various machine learning algorithms. Many countries and their environmental agencies in the world use the AQI for the real time spreading of the information on air quality. Although the basic concepts of air quality are similar, the practical implementations of each can differ. Applying AQIs on a common set of data can show large differences in the index values and concentration of pollutants

1.2 Problem Definition

Air pollution is a major public health problem that affects people all over the world. It can cause a variety of health problems, including respiratory infections, heart disease, and cancer. In order to address the problem of air pollution, it is important to have a better understanding of how air quality changes over time and space.

This project report describes the development of an IoT-based sensing system for monitoring and forecasting urban air quality. The system uses a network of sensors to collect data on air quality in real time. This data is transmitted to a central server, where it is analyzed and used to generate forecasts of air quality. The forecasts can then be used to inform public health decisions, such as school closures or advisories to stay indoors.

Following aspects will be discussed through out the project

- To identify effect of various environmental factors on air quality
- To recognize various trends in the air pollution levels over the course of a day.
- To identify effect of effect of humidity on air quality
- To identify effect of effect if temperature on air quality
- To study the dataset containing recordings of on field deployed air quality chemical sensor devices responses for more through study on the application og regression algorithms in this area

1.3 Machine learning

Machine learning is a subset of artificial intelligence (AI) and is focused on using various self-learning algorithms to derive knowledge from data in order to predict outcomes.

Machine learning is a field of study where models are built by training various learning algorithms. Machine learning algorithms are trained to build the models for identifying patterns in the data and make predictions. When trained, computer programs can take their own decisions and give outputs to the user. Machine learning is closely related to mathematical computations where algorithms perform various computations for predicting data. The efficiency of each algorithm is directly proportional to the amount of training data. Machine learning algorithms are basically categorized into supervised learning algorithms, unsupervised learning algorithms, and semi-supervised learning algorithms.

1.3.1 Supervised Learning Algorithms

Supervised learning algorithms are those algorithms where for every input data there is a output data mapped. The input data (basically a vector) always has a desired output value (signal). Supervised learning algorithms analyses the patterns between the data and develops a mapping function that can map any input to an output for new examples. There are various supervised learning algorithms such as Support Vector Machine (SVM), Decision tree, K-means, Naive Bayes, Random forest, and Artificial neural networks (ANN). [3]

1.3.2 Unsupervised Learning Algorithms

Unsupervised machine learning algorithms are the learning algorithms where, like the supervised learning algorithms, there isn’t any developing of a mapping function to map an input data to an output data. The main notion of unsupervised algorithms is to build representations of input data that can be used to predict future output, take decisions, and efficiently communicating the input to other machines. Few examples for unsupervised learning algorithms are K-means clustering, Principal Component analysis, and Hierarchical clustering. [4]

1.3.1 Semi-supervised Learning Algorithms

As the name suggests, semi supervised learning lies between supervised and unsupervised. Most of the semi-supervised learning algorithms are basically extensions of either supervised or unsupervised algorithms to include additional information of the other paradigms. In most of the tasks, there is a very small amount of labeled data because acquiring labels can be very difficult as the process requires human annotators, special devices, and slow experiments. Semi-supervised learning can be more effective than supervised learning because semi-supervised uses both labeled and unlabeled data for learning. In other words, the performance of semi-supervised learning is on par with that of supervised learning but with fewer labeled data. [5]

1.4 Regression

Regression is a supervised machine learning approach that is generally used to predict continuous values. Regression is majorly used for two purposes. First, it is used in forecasting and prediction of data, which are the applications of machine learning. Second, regression analysis is used to determine the relation between the dependent and independent variables in the data set. Linear regression, Ridge regression, and LASSO regression are a few examples for regression models. [6]

1.4.1 Linear Regression

Linear regression is a statistical modeling technique used to analyze the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship between the dependent variable and the independent variables.

In linear regression, the goal is to fit a linear equation to the data that best describes the relationship between the independent variables (also called predictors or features) and the dependent variable (also called the response or target variable). The linear equation is of the form:

$Y = \beta 0 + \beta 1X1 + \beta 2X2 + ... + \beta n*Xn + \varepsilon$

where:

- Y represents the dependent variable
- X1, X2, ..., Xn represent the independent variables
- β0, β1, β2, ..., βn are the coefficients of regression coefficients that quantify the relationship between the variables
- ε is the error term, representing the unexplained variability or noise in the model

The coefficients (β) in the linear equation are estimated using a method called ordinary least squares (OLS). The OLS method minimizes the sum of squared residuals (differences between the observed values and the predicted values). The optimal coefficients that minimize the sum of squared residuals are found, and these coefficients represent the best fit line or hyperplane in the case of multiple independent variables.

1.4.2 LASSO Regression

LASSO (Least Absolute Shrinkage and Selection Operator) regression is a popular linear regression method that incorporates a regularization term to perform both variable selection and regularization. It is particularly useful when dealing with high-dimensional datasets where the number of predictors (features) is much larger than the number of observations.

In LASSO regression, the objective is to minimize the sum of squared residuals (similar to ordinary least squares) while also penalizing the absolute values of the regression coefficients. This penalty term helps in shrinking some of the coefficients towards zero, effectively performing feature selection by setting some coefficients to exactly zero. This results in a sparse model where only a subset of predictors are selected as relevant.

LASSO regression has the advantage of performing both feature selection and regularization simultaneously, making it useful for situations where there are many predictors, and some of them may not be relevant. However, one limitation of LASSO is that it tends to arbitrarily select one among a group of correlated predictors, which may not always be desirable.

1.4.3 Decision Tree Regression

A decision tree is a supervised machine learning algorithm that is used for both classification and regression tasks. It is a flowchart-like structure where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents the outcome or class label.

To make predictions using a decision tree, you start at the root node and follow the path based on the values of the features until you reach a leaf node. The class label or predicted value associated with that leaf node is the output of the decision tree.

1.3 Project Overview

The project aims to develop an IoT-based sensing system for monitoring and forecasting urban air quality. The system utilizes Arduino UNO as the central microcontroller and incorporates various gas sensors such as MQ-135, MQ-2, and MQ-7. Additionally, a DHT11 sensor is employed to measure temperature and humidity values.

The primary objective of the project is to collect real-time data on air quality parameters. The gas sensors detect and measure the presence of specific air components, such as carbon dioxide, carbon monoxide, methane, and other pollutants. The DHT11 sensor captures temperature and humidity values, which are crucial factors in assessing air quality.

The collected sensor data is processed and transmitted using the IoT infrastructure, allowing for seamless connectivity and data transfer. The ESP-8266 Wi-Fi module facilitates the communication between the sensing system and a central data processing unit.

The project's ultimate goal is to create a comprehensive dataset consisting of various air components and environmental parameters. This dataset serves as a foundation for monitoring and forecasting urban air quality in real-time. By employing machine learning algorithms and predictive analytics techniques, the system can forecast future air quality conditions, enabling proactive measures to be taken for environmental management and improvement.

Overall, the project provides an innovative solution for monitoring and forecasting urban air quality using an IoT-based sensing system. By leveraging Arduino UNO, gas sensors, DHT11, and the ESP-32S Wi-Fi module, the system offers real-time data collection, analysis, and visualization, empowering decision-makers with valuable insights to enhance air quality in urban environments.

2. LITERATURE SURVEY

2.1 Literature Review

To analyze how machine learning helps in forecasting AQI and to identify the predictive algorithms for forecasting AQI we've used the Literature review research method.

• Firstly, we've identified keywords like air quality index, forecasting, prediction, supervised machine learning algorithms, machine learning which are mainly related to our thesis.

• Using the identified keywords, we searched for the previously done research related to prediction of AQI in google scholar, IEEE, etc.

• Some research works related to AQI prediction were gathered from the search.• Important research works were filtered by considering inclusion criteria and exclusion criteria.

Inclusion criteria:

1. Include studies that are related to both prediction of AQI and machine learning.

2. Peer assessment of the articles linked to AQI prediction.

3. Include the articles written in english.

4. Only published papers should be included.

Exclusion criteria:

1. The gathered works which are not scientific are excluded.

2. The works which don’t follow proper guidelines like without having an abstract are excluded.

2.2 Literature Review outcome

To understand and summarize how well machine learning helps in reducing air pollution and forecasting the AQI. To identify the best supervised machine learning algorithms to predict the AQI. We performed literature review and collected some previous research papers that helped us to achieve the above. The research papers were presented below.

1)In this paper [1], the authors have focused on the pollution caused by industrial activities. Here they have given a smart city model to monitor air pollution. They have established a wireless communication system through gas sensors on the busiest areas of the city. Where the sensors collect the pollution data from all the selected areas and store them in a central database. They have mainly used 3 sensors which detect up to 14 gases. Every local area sensor collects data and stores them in the local database and then sends them to the server/central database. They have given a chart about some gases maximum allowable emission limits and from that chart they have used the threshold value and gave the prediction about pollution.In conclusion they said that their aim was to aid in reducing respiratory problems due to industrial activities. The paper's main idea was to find out the increasing amount of harmful gases that caused air pollution and find particular solutions.

2)AQI Prediction using Machine Learning - A Review

In this research paper, the importance of machine learning in calculating AQI is explained and machine learning algorithms like linear regression, decision tree, random forest, artificial neural network, support vector machine were implemented to forecast AQI. Accuracy is considered to find the best fit model for the data [8].

3)Forecasting AQI using regression models: A case study on Delhi and Houston

In this research, SVR and linear models like multiple linear regression consisting of gradient descent, stochastic gradient descent, mini-batch gradient descent were implemented to predict the AQI. Among these models, SVR exhibited high performance in terms of investigated measures of quality [9].

4) Analyzing the impact of heating emissions on AQI based on principal component regression

In this paper a new method was proposed for investigating the quantitative impact of the heating emissions on the AQI and its assumptions and accuracy were verified as well [10].

By conducting the literature review and reading the above research papers we understood how machine learning helps in reducing the air pollution and calculating the AQI. By performing literature study on the above research papers, we identified that regression algorithms were the most usual supervised machine learning algorithms that are used to predict the AQI.

2.3 Flowchart

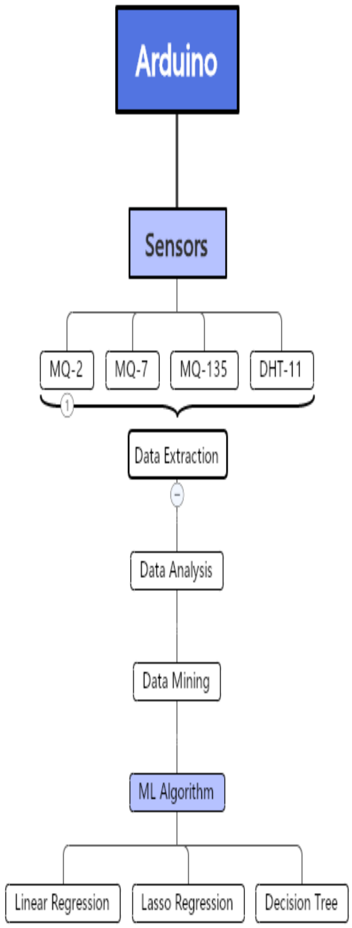


fig.1 flowchart

3. Methodology

3.1Requirement Specifications

3.1.1 Arduino UNO

Arduino Uno Rev 3 is a microcontroller board based on the ATmega328P [14]. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 26 analog inputs, a 16 MHz quartz crystal, 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.

Technical Specifications:

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB

Frequency (Clock Speed)	16 MHz
LED_BUILTIN	13
Length	68.6mm
Width	58.4mm
Weight	25g

table 1-Technical Specifications of Arduino

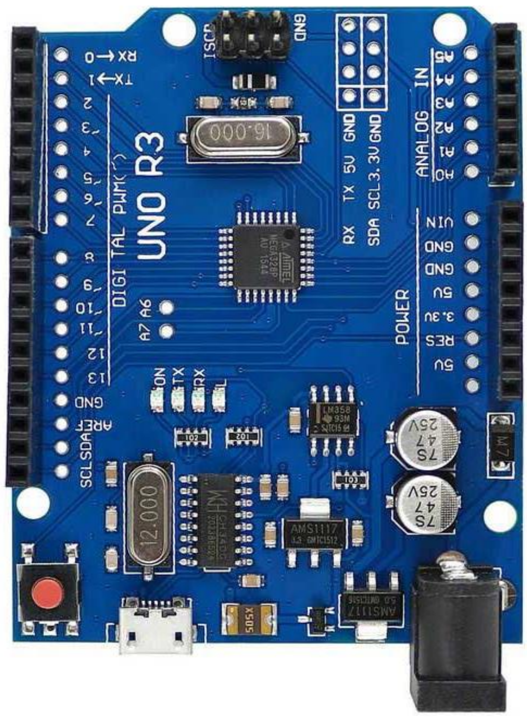
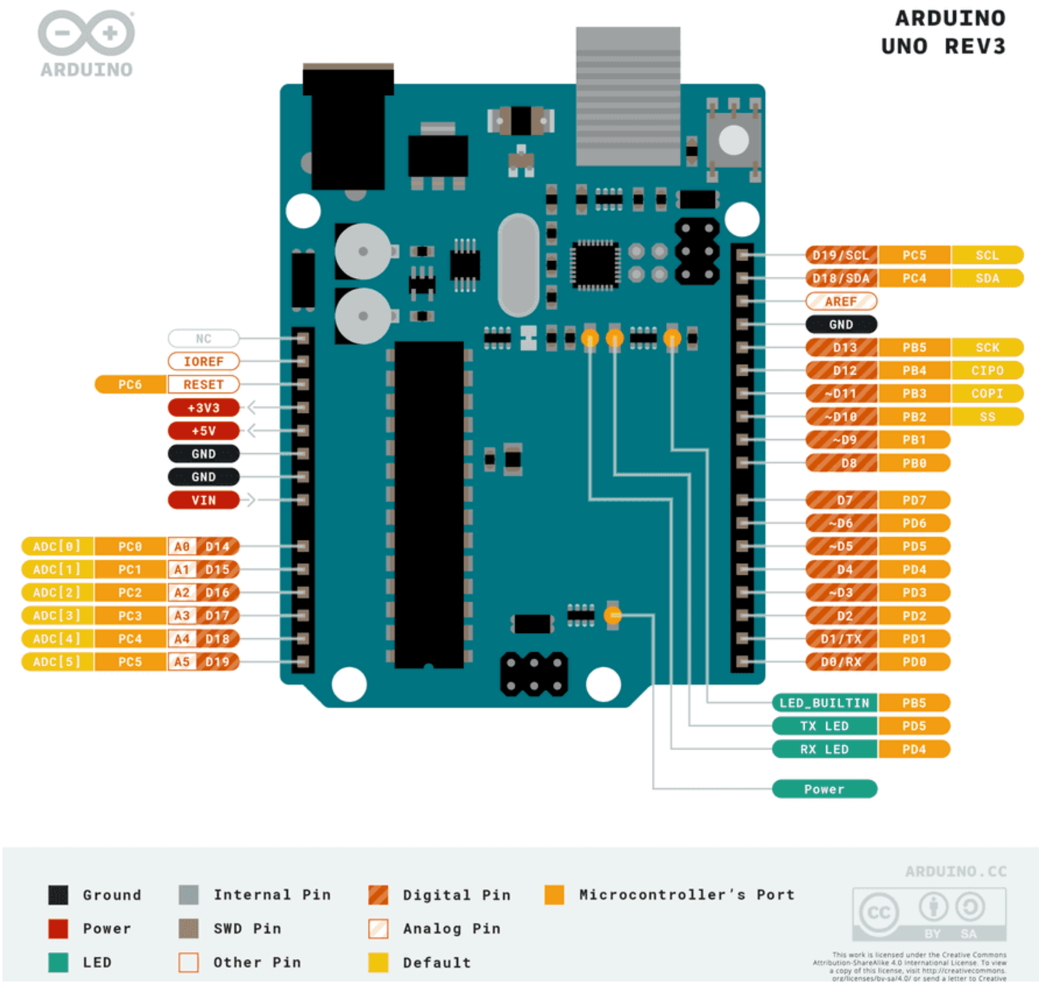


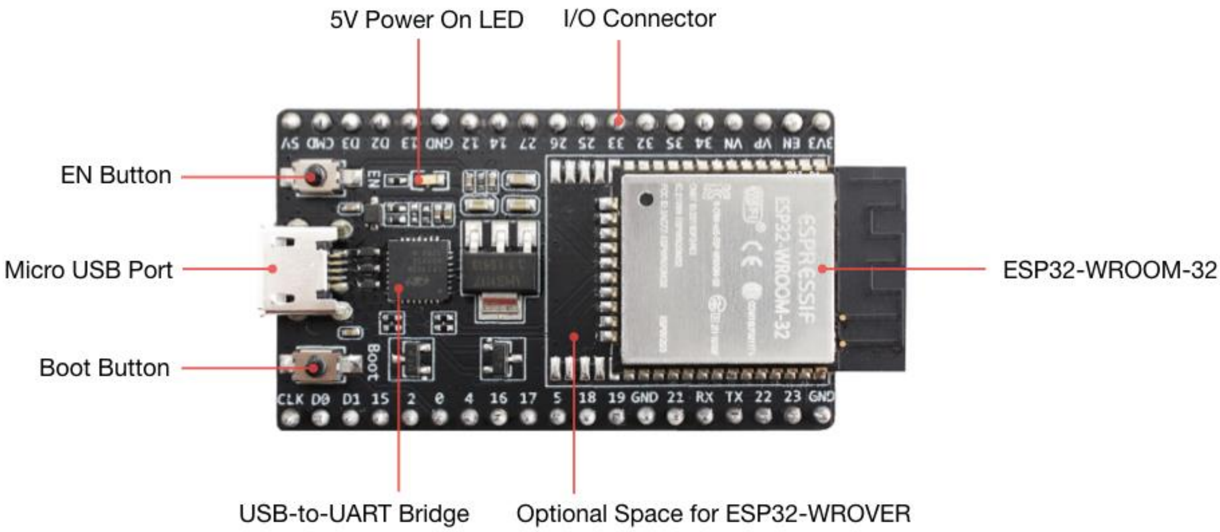
Fig 2.1- Arduino UNO



(Fig 2.2- Arduino pinout diagram)

3.1.2 ESP 32S

ESP32-S is a general-purpose WiFi-BT-BLE MCU module with powerful functions and a wide range of uses. It can be used for low-power sensor networks and demanding tasks, such as voice encoding, audio streaming, and MP3 decoding.



(Fig 3.1 -ESP32)

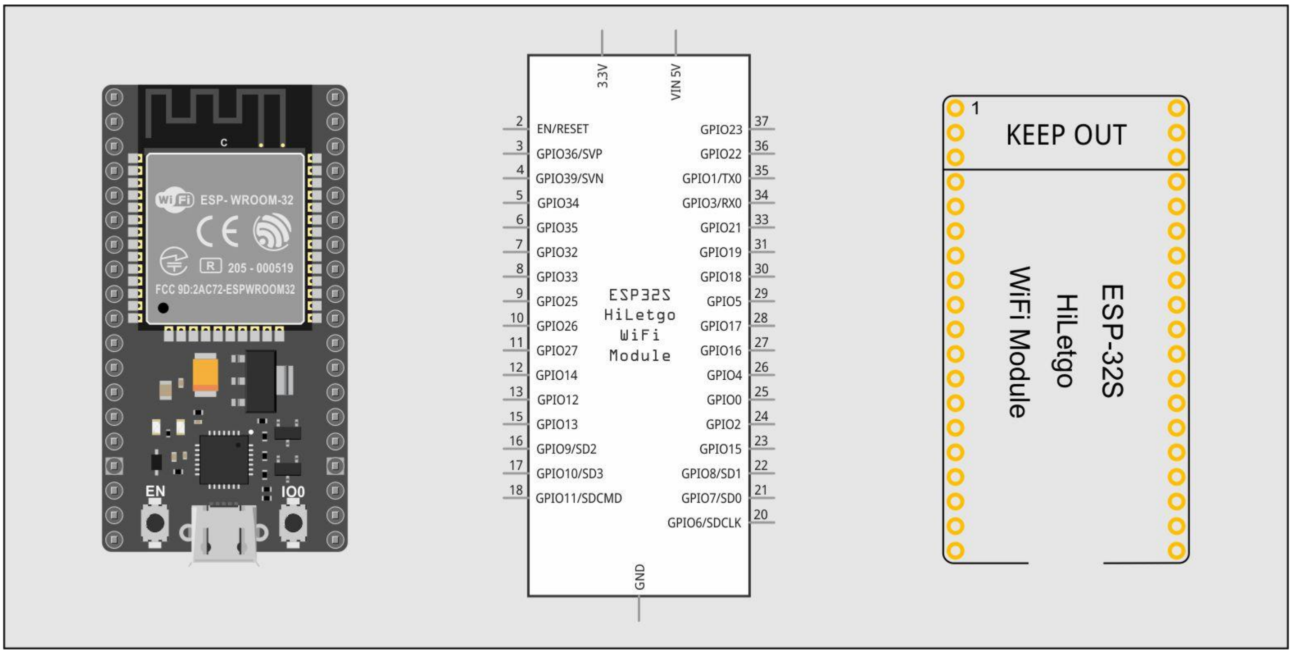


Fig 3.2-Esp 32S pinouts

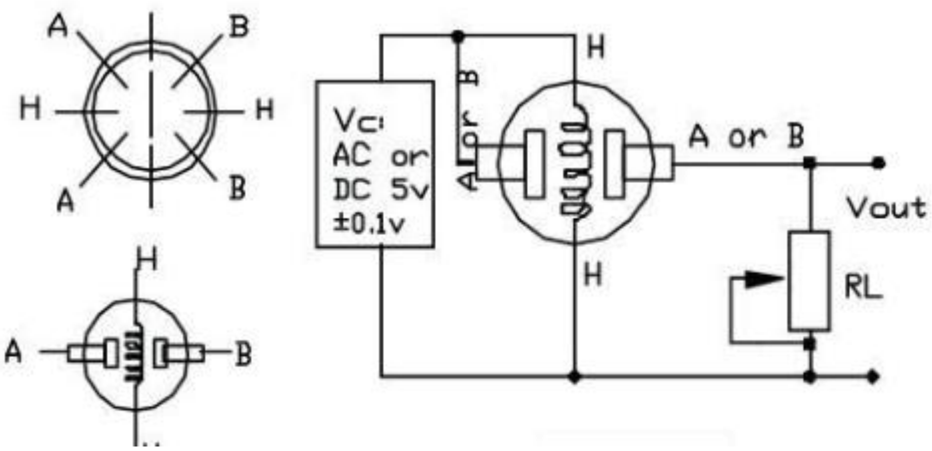
Technical Specification-

Microcontroller	Tensilica 32-bit RISC CPU Xtensa LX106
Operating Voltage	3.3V
Input Voltage	7-12V
Digital I/O Pins (DIO)	16
Analog Input Pins (ADC)	1
UARTs	1
SPIs	1
I2Cs	1
Flash Memory	4 MB
SRAM	64KB
Clock Speed	80 MHz

table -technical Specification ESP

3.1.3 Gas Sensors

For sensing toxic gases, we used MQ groove series gas sensors [15]. The MQ series of gas sensors use a small heater inside with an electro-chemical sensor. They are sensitive for a range of gases and are used indoors at room temperature. They can be calibrated more or less (see the section about "Load-resistor" and "Burn-in") but a known concentration of the measured gas or gasses is needed for that. The output is an analog signal and can be read with an analog input of the Arduino



(Fig 4.1- Circuit Design of MQ gas sensor)

a)MQ-2

The MQ2 sensor is one of the most widely used in the MQ sensor series. It is a MOS (Metal Oxide Semiconductor) sensor. Metal oxide sensors are also known as Chemiresistors because sensing is based on the change in resistance of the sensing material when exposed to gasses.

The MQ2 gas sensor operates on 5V DC and consumes approximately 800mW. It can detect LPG, Smoke, Alcohol, Propane, Hydrogen, Methane and Carbon Monoxide concentrations ranging from 200 to 10000 ppm.

MQ-2 module is suitable for detecting H2, LPG, CH4, CO, Alcohol, Smoke or Propane. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible.

(Fig 4.2- MQ-2 Sensor)

(Fig 4.3- MQ2 Gas Sensor Module Pinout)

Technical Specifications:

Boost chip	PT1301
Power	2.5V ~ 5.0V
Load resistance	20KΩ
Heating resistance	33Ω ± 5%
Heating consumption	<800mw
Sensing Resistance	10 KΩ – 60 KΩ
Concentration Range	200 – 10000 ppm
Dimensions	3.6 x 2 x 1 cms
Weight	5 grams

table 3.1 -Technical Specifications of MQ2

Pin outs:

PIN	Description
DOUT	Digital data output
AOUT	Analog data output
GND	Ground
VCC	Power input (2.5V~5.0V)

table 3.2-Pinouts of MQ2

b) MQ-7

The gas sensing material used in the MQ-7 gas sensor is tin dioxide (SnO_2), which has low conductivity in clean air. When carbon monoxide gas exists in the environment where the sensor is located, the conductivity of the sensor increases with the increase of carbon monoxide gas concentration in the air.

Fig 4.4- MQ-7 Sensor

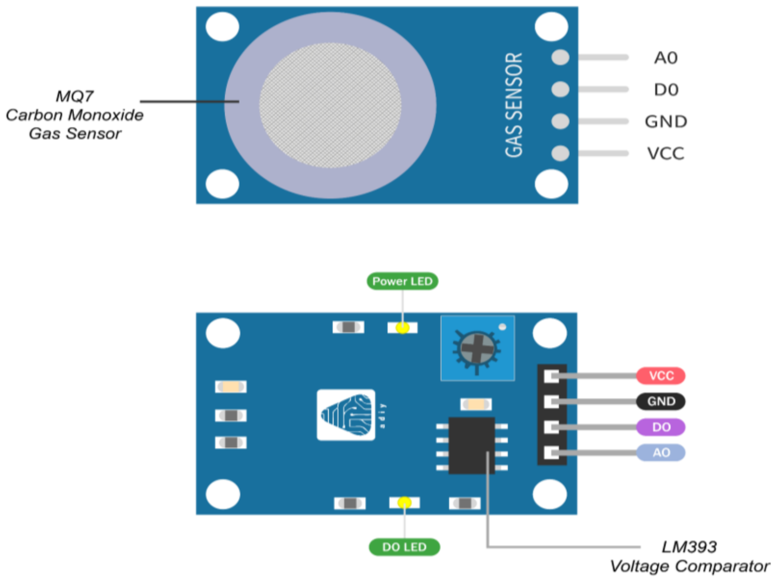


Fig 4.5- MQ7 Gas Sensor Module Pinout

Technical Specifications:

Boost chip	PT1301
Power	2.5V ~ 5.0V
Load resistance	10KΩ
Heating resistance	33Ω ± 5%
Heating consumption	About 350mW

Sensing Resistance	2 KΩ – 20 KΩ
Concentration Range	20- 2000 ppm
Dimensions	3.5 x 2 x 1.1 cms
Weight	5 gr

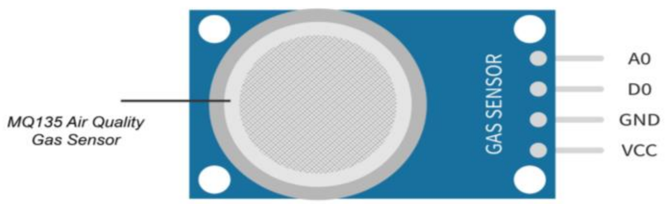
Pin outs

Pin No	Symbol	Description
1	DOUT	Digital data output
2	AOUT	Analog data output
3	GND	Ground
4	VCC	Power input (2.5V ~ 5.0V)

table 4 -Technical Specifications of MQ7

c)MQ135

MQ-135 gas sensor applies SnO2 which has a lower conductivity in the clear air as a gas-sensing material. In an atmosphere where there may be polluting gas, the conductivity of the gas sensor rises along with the concentration of the polluting gas increases. MQ-135 performs a good detection of smoke and other harmful gases, especially sensitive to ammonia, sulfide and benzene steam. Its ability to detect various harmful gases and lower cost make MQ-135 an ideal choice of different applications of gas detection.



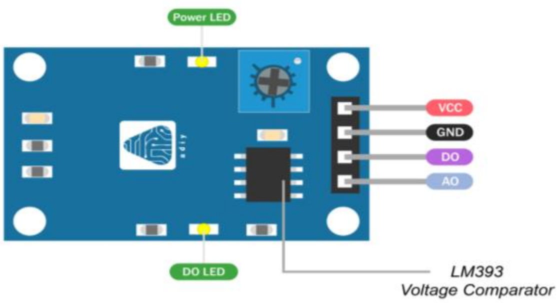


Fig 4.6- MQ-135 Sensor

Fig 4.7-MQ135 Gas Sensor Module Pinout

Technical Specifications:

Boost chip	PT1301
Power	5.0V
Load resistance	Can adjust
Heating resistance	33Ω ± 5%
Heating consumption	< 800mW
Sensing Resistance	30KΩ-200KΩ
Concentration Range	10 – 1000 ppm
Dimensions	3.6 x 2.4 x 2.5 cms
Weight	11 grams

Pin outs

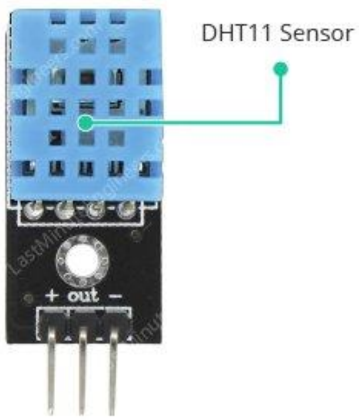
Pin No	Symbol	Description
1	DOUT	Digital data output
2	AOUT	Analog data output
3	GND	Ground
4	VCC	Power input (2.5V ~ 5.0V)

table 5 - Technical Specifications of MQ 135

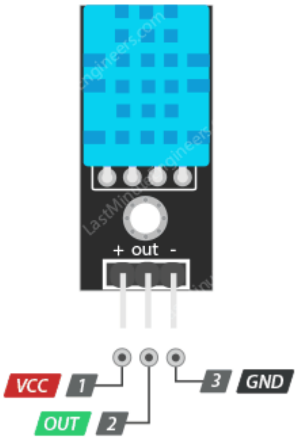
3.1.4 DHT-11 Sensor

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data. You can get new data from it once every 2 seconds, so when using the library from Adafruit, sensor readings can be up to 2 seconds old.

DHT11 sensors typically require an external 10K pull-up resistor on the output pin for proper communication between the sensor and the Arduino. However, because the module already includes a pull-up resistor, no need to add one. The module also includes a decoupling capacitor for filtering power supply noise.



(Fig 5.1- DHT11 sensor)



(Fig 5.2- DHT11 sensor pinouts)

Technical Specifications-

Temperature range	0-50 ± 2 °C
Humidity range	20-80 ± 5%
Sampling rate	1 Hz
Dimensions	1.5cm x 1.2cm x 0.5cm

Pinouts-

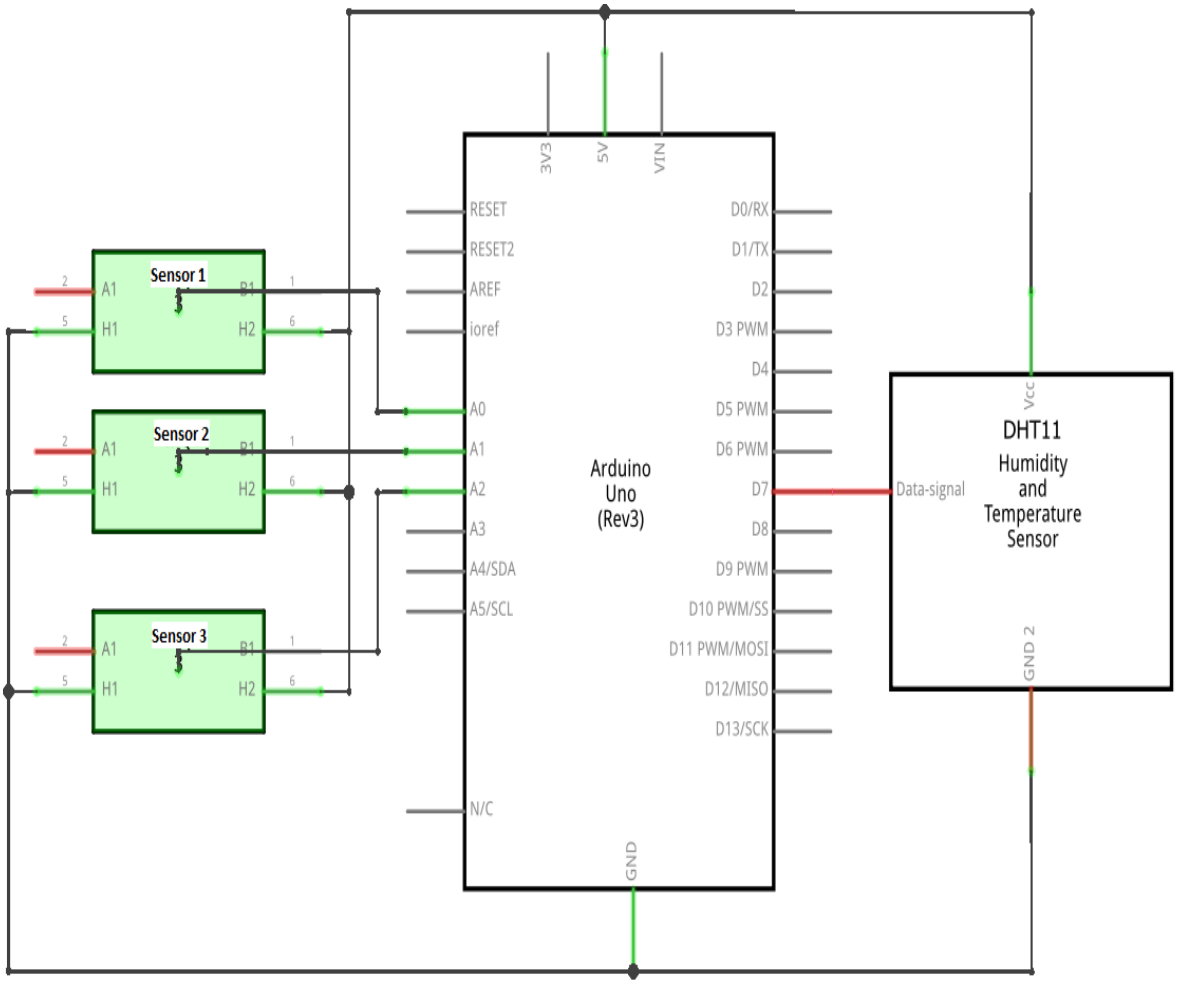
Pin No	Symbol	Description
1	VCC	Power input (3.3V ~ 5.0V)
2	OUT	output
3	GND	Ground

table 6- Technical Specifications of DHT11

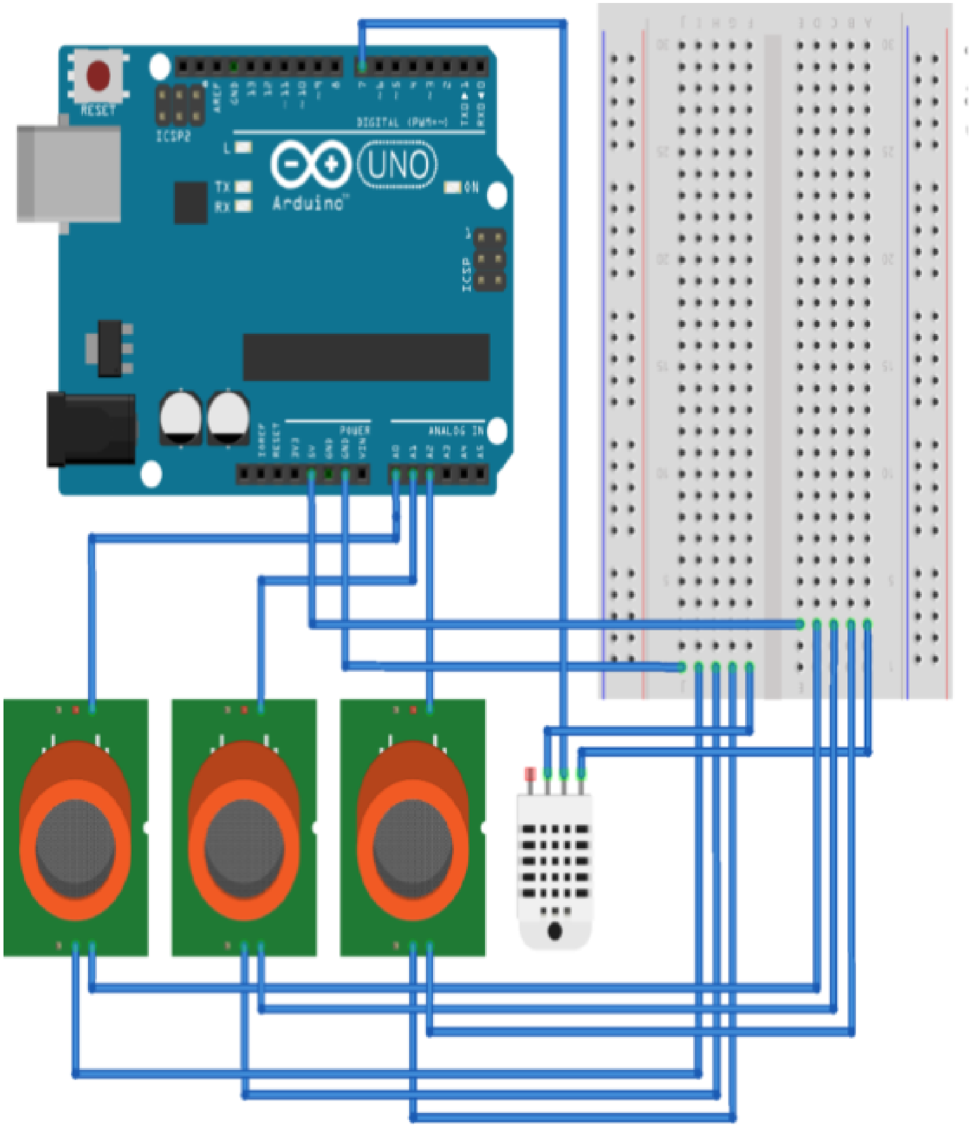
3.2 Experimentation

3.2.1 Module Connection

To measure gas concentration in the air, we used Arduino microcontroller and 3 different gas sensors. The gas sensors are sensitive to particular gases. There is only one analog voltage we can get from the sensor.



(Fig 6.1- Pin Diagram for our device)



(Fig 6.2- Circuit Diagram of our device)

We used Arduino ethernet shield to upload sensor data in a cloud platform named

3.2.2 Environment Setup

To connect the whole device and make it functional, we need to connect the Arduino with a USB cable with a computer. Coding for the gas sensors and conditional decision making, we used Arduino Sketch IDE. After preparing the code, it can be compiled and uploaded to the Arduino. Before uploading, the proper Arduino board and COM port has to be selected from the menu of the IDE.

(Fig- Actual look of our device)

The Ethernet Shield which is stacked on the Arduino is connected to a router using a RJ45 straight-through cable. Using the router's internet connection, it will upload sensor data to a specific channel of ThingSpeak cloud platform. The channel Id and conditions for uploading to could are maintained inside the code.

After uploading the code, it takes several minutes to calibrate all the sensors. Then it starts to show the gas concentration in the serial port monitor of the IDE.

3.3 Building Dataset:

All the data taken by the Arduino from the gas sensors are exported automatically in a particular format to an Excel Sheet which will be used later on to analyze. With big amount of data and years of data, we can have the advantage to find more accurate results. Also we can predict the air quality for future days.

4.Analysis with related dataset

4.1 Introduction:

In this chapter, we'll do some analysis with an old dataset. We will discuss about it and the parameters and types of data it has. And we will also try to measure accuracy of the dataset so that with future data retrieved with our device, how we can predict town pollution.

4.2 Dataset information:

The dataset was obtained from the website of Machine Learning Repository of UCI (University of california, Irvine)[13]. The dataset contains the responses of a gas multisensor device deployed on the field in an Italian city. Hourly responses averages are recorded along with gas concentrations references from a certified analyzer.

The dataset contains 9358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multi Sensor Device. The device was located on the field in a significantly polluted area, at road level,within an Italian city. Data were recorded from March 2004 to February 2005 (one year)representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. Ground Truth hourly averaged concentrations for CO, Non Methane Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx) and Nitrogen Dioxide (NO2) and were provided by a co-located reference certified analyzer. Evidence of cross-sensitivities as well as both concept and sensor drifts are present as described in De Vito et al., Sens. And Act. B, Vol. 129,2,2008 (citation required) eventually affecting sensor concentration estimation capabilities. Missing values are tagged with -200 values.

This dataset can be used exclusively for research purposes. Commercial purposes are fully exclude

Attribute Information:

0 Date (DD/MM/YYYY)
1 Time (HH.MM.SS)
2 True hourly averaged concentration CO in mg/m^3 (reference analyzer)
3 PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)
4 True hourly averaged overall Non Methane HydroCarbons concentration in micro g/m^3 (reference analyzer)
5 True hourly averaged Benzene concentration in microg/m^3 (reference analyzer)
6 PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)
7 True hourly averaged NOx concentration in ppb (reference analyzer)

8 PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)
9 True hourly averaged NO2 concentration in microg/m^3 (reference analyzer)
10 PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)
11 PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)
12 Temperature in Â°C
13 Relative Humidity (%)
14 AH Absolute Humidity

4.3 Analysis

4.3.1 Environment Setup:

For our analysis we used google colab. We installed several libraries required for the analysis. We tried to find the accuracy of the dataset by using various regression models.

To implement experimentation we've used the following technologies.

- **import pandas as pd:** This line imports the pandas library and assigns it the alias 'pd'. Pandas is a powerful library for data manipulation and analysis.
- **import numpy as np:** This line imports the numpy library and assigns it the alias 'np'. NumPy is a fundamental package for scientific computing with Python and provides support for large, multi-dimensional arrays and matrices.
- **import matplotlib.pyplot as plt:** This line imports the pyplot module from the matplotlib library and assigns it the alias 'plt'. Matplotlib is a plotting library for creating static, animated, and interactive visualizations in Python.
- **%matplotlib inline:** This is a magic command for Jupyter Notebook, which allows plots to be displayed directly in the notebook.
- **import seaborn as sns:** This line imports the seaborn library and assigns it the alias 'sns'. Seaborn is a data visualization library built on top of Matplotlib, providing a high-level interface for creating attractive and informative statistical graphics.
- **import tensorflow as tf:** This line imports the TensorFlow library, which is an open-source machine learning framework. TensorFlow provides a wide range of tools and resources for building and deploying machine learning models.
- **from tensorflow.python.keras.models import Sequential:** This line imports the Sequential class from the Keras models module. Keras is a high-level neural networks API written in Python and is a part of TensorFlow.
- **from tensorflow.python.keras.layers import Dense:** This line imports the Dense class from the Keras layers module. Dense is a fully connected layer, which is a common type of layer used in neural networks.
- **from tensorflow.python.keras.wrappers.scikit_learn import KerasRegressor:** This line imports the KerasRegressor class from the Keras wrappers module. KerasRegressor is a wrapper for the Keras models that provides compatibility with scikit-learn.

- **from sklearn.model_selection import KFold:** This line imports the KFold class from the scikit-learn library. KFold is a cross-validation iterator that splits the dataset into train and test sets for multiple iterations.
- **from sklearn.model_selection import cross_val_score:** This line imports the cross_val_score function from the scikit-learn library. cross_val_score is used to evaluate the performance of a machine learning model using cross-validation.
- **from tensorflow.python.keras.layers.core import Dense, Activation:** This line imports the Dense and Activation classes from the Keras core module. Dense is a fully connected layer, and Activation represents an activation function applied to the layer.

5.RESULTS

- **Linear Regression**

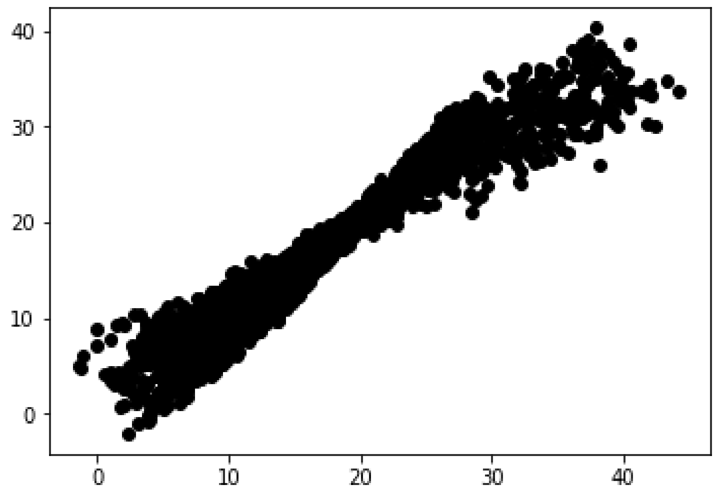


fig 7.1- linear regression

- **Data Analysis**

the updated air_quality DataFrame after performing the previous operations. It allows you to see the modified DataFrame with missing values removed and -200 values replaced with NaN.

Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)
18.00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	1268.0
19.00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	972.0
20.00.00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555.0	1074.0
21.00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584.0	1203.0
22.00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	1110.0
23.00.00	1.2	1197.0	38.0	4.7	750.0	89.0	1337.0	96.0	1393.0	949.0
00.00.00	1.2	1185.0	31.0	3.6	690.0	62.0	1462.0	77.0	1333.0	733.0
01.00.00	1.0	1136.0	31.0	3.3	672.0	62.0	1453.0	76.0	1333.0	730.0
02.00.00	0.9	1094.0	24.0	2.3	609.0	45.0	1579.0	60.0	1276.0	620.0
03.00.00	0.6	1010.0	19.0	1.7	561.0	NaN	1705.0	NaN	1235.0	501.0
04.00.00	NaN	1011.0	14.0	1.3	527.0	21.0	1818.0	34.0	1197.0	445.0
05.00.00	0.7	1066.0	8.0	1.1	512.0	16.0	1918.0	28.0	1182.0	422.0
06.00.00	0.7	1052.0	16.0	1.6	553.0	34.0	1738.0	48.0	1221.0	472.0
07.00.00	1.1	1144.0	29.0	3.2	667.0	98.0	1490.0	82.0	1339.0	730.0
08.00.00	2.0	1333.0	64.0	8.0	900.0	174.0	1136.0	112.0	1517.0	1102.0
09.00.00	2.2	1351.0	87.0	9.5	960.0	129.0	1079.0	101.0	1583.0	1028.0
10.00.00	1.7	1233.0	77.0	6.3	827.0	112.0	1218.0	98.0	1446.0	860.0
11.00.00	1.5	1179.0	43.0	5.0	762.0	95.0	1328.0	92.0	1362.0	671.0
12.00.00	1.6	1236.0	61.0	5.2	774.0	104.0	1301.0	95.0	1401.0	664.0

fig. 7.2-data analysis

6.Conclusion and future work

6.1 Conclusion

Here, we tried to create a dataset which can also be used for future work for further analysis we used a dataset which has year long data included in it

6.2 Limitations and Shortcomings

- Since the sensors were cheap, they have certain limitations, such as lower accuracy, shorter lifespan, and potential variability in performance. However, when used in combination with other data sources and appropriately calibrated, these sensors can still provide valuable insights into urban air quality, especially when deployed in a large-scale network.

6.3 Advantages

1. Affordability: Cheap sensors are cost-effective, making them accessible to a wider range of users, including individuals, communities, and organizations with limited budgets. This affordability enables the deployment of a larger number of sensors across a city, resulting in a denser and more comprehensive air quality monitoring network.

2. Scalability: With inexpensive sensors, it becomes feasible to deploy a large-scale sensing system across urban areas. This scalability allows for widespread coverage, enabling monitoring and forecasting of air quality in various neighborhoods and regions within a city.

3. Accessibility: Cheap sensors democratize access to air quality information. By employing low-cost devices, citizens can actively participate in monitoring and contribute to improving air quality in their communities. Increased accessibility fosters public engagement, awareness, and collaboration towards addressing air pollution challenges.

4. Rapid deployment: Due to their lower cost, it is easier and faster to deploy a network of cheap sensors. This advantage is particularly useful in situations that require immediate action, such as responding to sudden pollution events or identifying pollution hotspots in real-time.

5. Data density: Deploying a larger number of cheap sensors increases the density of data points, resulting in a more granular understanding of air quality conditions. This finer resolution facilitates the identification of localized pollution sources, the mapping of

pollution patterns across different areas, and the detection of spatial variations in air quality within a city.

6. Citizen science: Cheap sensors encourage citizen science initiatives, empowering individuals and communities to actively contribute to environmental monitoring efforts. By involving the public, these initiatives can gather valuable data from diverse locations, thus enhancing the overall accuracy and coverage of air quality monitoring systems.

7. Data fusion and calibration: Although cheap sensors may have lower accuracy compared to more expensive counterparts, their data can be combined with data from other sources, such as official monitoring stations or satellite observations. Through data fusion and calibration techniques, the information collected by inexpensive sensors can be validated and adjusted, improving the overall accuracy and reliability of air quality forecasts.

6.3 Future Scope

In this project even though we have implemented Arduino setup the data used was static for more detailed study regarding the regression that means the data will be fixed and it remains the same after it’s collected. However the government updates the data hourly. Further we can use real-time data analysis using cloud to obtain better outcomes for greater performance as the data updates for every particular interval of time. We can further ensemble two or more machine learning algorithms and process large data to get more accurate results.

Appendix A: List of Acronyms

1. AQI: Air Quality Index
2. IoT: Internet of Things
3. PM: Particulate Matter
4. CO: Carbon Monoxide
5. NO2: Nitrogen Dioxide
6. SO2: Sulfur Dioxide
7. O3: Ozone

8. VOCs: Volatile Organic Compounds
9. RH: Relative Humidity
10. Temp: Temperature
11. ML: Machine Learning
12. H2: Hydrogen
13. CH4: Methane
14. AQMP: Air Quality Management Project
15. LPG: Liquefied petroleum gas
16. EPA: Environmental Protection Agency
17. LPWA: Low power wide area
18. NaN: Not a number

Appendix B: Dataset for classification

*Only a few of 9358 instances were presented

6. REFERENCES

[1]A. Alshamsi, Y. Anwar, M. Almulla, M. Aldohoori, N. Hamad and M.Awad, "Monitoring pollution: Applying IoT to create a smart environment," 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, 2017, pp. 1-4.

doi: 10.1109/ICECTA.2017.8251998
URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8251998&isnumber=8251917
[2]A. Plaia and M. Ruggieri, “Air quality indices: a review,” Reviews in Environmental Science and Bio/Technology, vol. 10, no. 2, pp. 165–179, 2011.
[3]M. Somvanshi, P. Chavan, S. Tambade, and S. Shinde, “A review of machine learning techniques using decision trees and support vector machines,” in 2016 international conference on computing communication control and automation (ICCUBEA). IEEE, 2016, pp. 1–7.
[4] Z. Ghahramani, “Unsupervised learning,” in Summer school on machine learning. Springer, 2003, pp. 72–112
[5] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” Synthesis lectures on artificial intelligence and machine learning, vol. 3, no. 1, pp. 1–130, 2009.
[6] D. Maulud and A. M. Abdulazeez, “A review on linear regression comprehensive in machine learning,” Journal of Applied Science and Technology Trends, vol. 1, no. 4, pp. 140–147, 2020.
[7] X. Su, X. Yan, and C.-L. Tsai, “Linear regression,” Wiley Interdisciplinary Reviews: Computational Statistics, vol. 4, no. 3, pp. 275–294, 2012.
[8] T. Madan, S. Sagar, and D. Virmani, “Air quality prediction using machine learning algorithms –a review,” in 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN),2020, pp. 140–145.
[9]S. S. Ganesh, S. H. Modali, S. R. Palreddy, and P. Arulmozhivarman, “Forecasting air quality index using regression models: A case study on delhi and houston,” in 2017 International Conference on Trends in Electronics and Informatics (ICEI), 2017, pp. 248–254.
[10]H. Li, S. You, H. Zhang, W. Zheng, W.-l. Lee, T. Ye, and L. Zou, “Analyzing the impact of heating emissions on air quality index based on principal component regression,” Journal of cleaner production, vol. 171, pp. 1577–1592, 2018.
[11] V. S. Esther Pushpam, N. S. Kavitha, and A. G. Karthik, “IoT Enabled Machine Learning for Vehicular Air Pollution Monitoring,” 2019 Int. Conf. Comput. Commun. Informatics, ICCCI 2019, pp. 1–7, 2019, doi: 10.1109/ICCCI.2019.8822001.
[12]J. Ranstam and J. Cook, “Lasso regression,” Journal of British Surgery, vol. 105, no. 10, pp. 1348–1348, 2018.
[13]Dataset link: https://archive.ics.uci.edu/ml/datasets/air+quality
[14]