CHAPTER 1

INTRODUCTION

1.1 GENERAL BACKGROUND

Wireless sensor networks (WSNs) are progressively affecting everyday living. A WSN is a network consisting of sensor nodes. Each sensor can detect certain factors like air pressure, air composition, and water quality. WSNs are used in a wide variety of settings, including personal space, industrial floors, agriculture, home utility monitoring systems, factory automation, automotive, and many other fields. WSNs are related to the concept of IoT. In IoT, devices are interconnected to transmit data via distributed sensor networks. IOT has useful applications in the medical field. Devices such as smartphones and sensing systems can be associated to create an infrastructure that provides access to health care information and services. This approach is referred to as "Mobile-Health". Mobile-Health can be viewed as the consequence of the convergence of wireless communication systems, WSNs, and global computing tools.

Governments and Citizens are looking for scientific intellect to challenge the common threat of pollution in its many procedures. Currently mobile apps are able to accomplish functions like reporting status of air quality, air quality forecasts, air quality monitoring in a particular area, and risks highlighting connected with threshold breaking quality, etc. There are also mobile apps designed for mass polluting sectors like industries. Industries or corporate organizations are now able to integrate and streamline environmental processes, including air emissions analyses, water and energy management, and waste reduction specific to them, through such apps.

Around 90% of the populations in low and center wage nations are presented to perilous levels of encompassing air contamination. The World Bank works with creating nations and advancement accomplices to diminish contamination by supporting checking and examination, administrative changes, and ventures. In 2016 for instance, the Bank conferred US$1 billion to enable China to enhance air quality by lessening discharges of particular air poisons from mechanical, transport and country sources in the territory of Hebei, and by expanding vitality effectiveness and clean vitality through imaginative financing in the Beijing-Tianjin-Hebei district that covers the capital region and neighboring regions.

Death associated with encompassing air contamination have expanded in intensely populated, quick urbanizing areas, while death identified with cooking and warming homes with strong energizes have stayed consistent in spite of advancement additions and upgrades in wellbeing administrations. Illnesses credited to the two sorts of air contamination caused 1 of every 10 deaths in, at least 2013 than six times the quantity of death caused by intestinal sickness.

The Cost of Air Pollution: Strengthening the financial case for activity, a joint investigation of the World Bank and the Institute for Health Metrics and Evaluation (IHME), looks to appraise the expenses of unexpected losses identified with air contamination, to fortify the case for activity and encourage basic leadership with regards to rare assets. An evaluated 5.5 million lives were lost in 2013 to illnesses related with outside and family unit air contamination, causing human enduring and decreasing financial improvement.

While pollution-related deaths strike mainly young children and the elderly, premature deaths also result in lost labour income for working-age men and women. The report finds that annual labour income losses cost the equivalent of almost 1 percent – 0.83 percent -- of Gross Domestic Product (GDP) in South Asia. In East Asia and the Pacific, where the population is ageing, labour income losses represent 0.25 percent of GDP, while in Sub Saharan Africa, where air pollution impairs the earning potential of younger populations, annual labour income losses represent the equivalent of 0.61 percent of GDP. Every year 3 million deaths occur as a result of exposure to ambient (outdoor) air pollution. Every year 4.3 million deaths as a result of household exposure to smoke from fuels and dirty cook stoves. The world's population lives in places where air quality exceeds WHO guideline limits of 92%.

1.2 OBJECTIVES

The air quality monitoring program design will be dependent upon the monitoring specific objectives specified for the air quality management in the selected area of interest. There might be different objectives for the development of the environmental monitoring and surveillance system. Normally, the system will have to provide on-line data and information transfer with a direct /automatically/ on-line quality control of the collected data. Several monitors, sensors and data collection systems may be applied to make on-line data transfer and control possible.

The main objectives stated for the development of an air quality measurement are: −

- monitor current levels as a baseline for assessment,
- check the air quality relative to standards or limit values,
- determine the concentration of pollutants in different locations
- enable comparison of the air quality data from different areas and countries,
- Predict the future pollution level

1.3 SCOPE

Air pollution monitoring is an important application of Internet of Things. In our project we propose an air pollution monitoring system using IoT. The main objective of this model is to monitor and analyze the air pollution from any location. The proposed system monitors the air pollutants from different location and uploads the data to the cloud server for further processing and analysis. The hardware setup of the proposed system is detailed in our project. Real time deployment of the proposed model is to be carried out in the future. Maintenance of the equipment in all weather conditions, transmission of data effectively etc. are the challenges that need to be addressed.

1.4 SCHEME

Here to make an IOT Based Air Pollution Monitoring System in which we will monitor the Air quality over an Android application using internet and will warn the users when the air quality goes down beyond a certain level, means when there is sufficient amount of harmful gases are present in the air like CO, smoke, methane and NH3. It will show the air quality in PPM on the application so that we can monitor it very easily. In our IOT project, you can monitor the pollution level from anywhere using your mobile. We can install this system anywhere and can also trigger some device when pollution goes beyond some level, like we can send alert message to the user.

The rest of the report is organized as follows. Chapter 2 gives a description about various works done in the area of air pollution monitoring system. Chapter 3 describes the overview of existing system and proposed system. Chapter 4 presents the implementation details of the proposed work. Chapter 5 describes the system implementation. Chapter 6 describes experimental results and analysis. Conclusion of the work and the future research directions in this area is provided in Chapter 7.

CHAPTER 2

LITERATURE SURVEY

Air pollution in large urban areas has a drastic effect on humans and the environment. Ecological issues in India are growing quickly. Air contamination is mainly caused by vehicles and industries which cause various respiratory diseases such as asthma and sinusitis. The quality of air is inferior in metropolitan cities like Kolkata, Delhi, and Mumbai due to a large amount of carbon dioxide and other harmful gases emitted from vehicles and industries.

An extensive number of projects have been described in the literature that utilize low-cost air pollution sensing devices that can be carried by individuals or by versatile vehicles [3]. In two studies, [4] [5], the authors demonstrated an environmental sensing approach that reinvigorate attention and sympathy of citizens toward pollution. Exposure Sense [6] is a portable participatory sensing framework that is used to screen one's everyday activities.

In another study [7], the authors present a cloud-based system that uses knowledge-based discovery to find real-time air quality data. The data are collected by monitoring stations that are placed in various geo-locations. This system uses mobile clients for monitoring purposes.

Re et al. [8] presented an Android application which provides users with information about air quality. By joining user area information and metropolitan air quality data provided by monitoring stations, this application provides a ubiquitous and unobtrusive monitoring [9] framework that is ready to advise users about their daily air pollution exposure.

Reshi et al. [10] designed a WSN platform, called VehNode that provided automobiles with the capacity to monitor the level of pollutants in smoke released by the vehicle.

Mujawar e al. [11] outlined an air pollution contamination measuring system utilizing WSN for use in Solapur City. Micro sensor nodes detect the target gas by measuring the sensing layer's electrical conductivity. When the gases touch the surface of the sensor they are assimilated and the conductivity changes. Also, a semiconductor sensor is used at the emission outlet of the vehicle to sense the level of pollutants and transmit this level to the microcontroller [2].

In another study by De Nazelle et al. [12], the authors demonstrated environmental sensing approaches that reinvigorated the awareness and sympathy of individuals towards pollution.

“Air pollution is a challenge that threatens basic human welfare, damages natural and physical capital, and constrains economic growth. We hope this study will translate the cost of premature deaths into an economic language that resonates with policy makers so that more resources will be devoted to improving air quality. By supporting healthier cities [13] and investments in cleaner sources of energy, we can reduce dangerous emissions, slow climate change, and most importantly save lives,” said Laura Tuck, Vice President for Sustainable Development at the World Bank.

The report of air pollution is a burden of disease associated to urgent call the government for necessary action,” said Dr. Chris Murray, Director of IHME. “Of all the different risk factors for premature deaths, this is one area, the air we breathe, over which individuals have little control. Policy makers in health and environment agencies, as well as leaders in various industries, are facing growing demands – and expectations – to address this problem.”

2.1 Wireless Sensor Network

A WSN is an organization of sensor nodes, which gather data using wireless technology. WSNs have limited computation power and memory. WSNs are used to monitor low-frequency data at remote locations.

The sensor network layout is intended to recognize all the gases associated with pollution in a given location. In our project, NH_3, CO, and CH_4 gases [1] are considered as parameters of air quality. Unlike the customary wired framework, expenses of WSN would be minimal. By including more devices, the network can be scaled without much re-work or complex modifications. Moreover, WSN can powerfully adjust to evolving situations.

2.2 Internet of Things

Internet of Things (IoT) [14] [15] encompasses regular objects (“Things”) that have network availability, permitting them to send and get information. "Things" include people, information, software agents, or any other virtual participating actors. There are five kinds of "Things" utilized in our project: Arduino Uno [16] (Microcontroller), ESP8266 (Wi-Fi Module), Cloud service, gas sensors and Android. These "Things" are coordinated to make a framework such that each "Thing" can work individually and can gather, store, and recover information to address the problem.

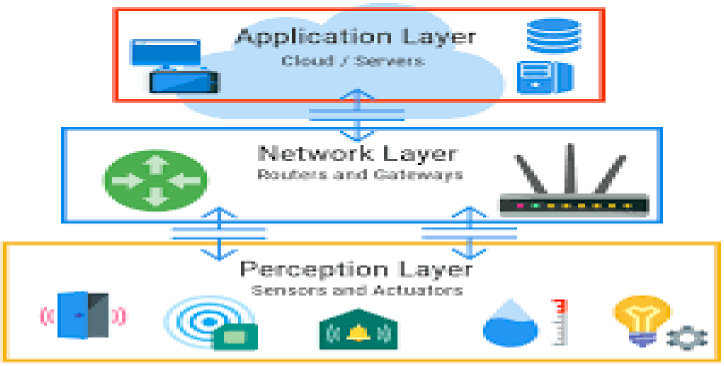

Fig.2.1. Layered architecture of IoT

The fundamental IoT model is a three-layered design composed of the Application, Network, and Sensor Layers (Figure 1). In the Sensor Layer, data is attained from the real world. The Network Layer receives data sent by the Sensor Layer. This layer acts as middleware between sensor layer and application layer. Lastly, the Application Layer provides services to incorporate or investigate the data that has been received from the previous two layers. In the work, a 3-layered architecture was used.

2.3 Global Positioning System (GPS)

GPS sends data to receivers that are installed on the ground. Data is collected from satellites orbiting around the earth. Anyone can utilize GPS for free with a suitable receiver. This technology is used for navigation systems. Google Maps Navigation was utilized as a part of this project. Google Maps Navigation is a versatile application that has been incorporated into the Google Maps mobile application. Google Maps Navigation utilizes an internet connection to access the GPS navigation system to find the user's location. The user can enter a destination into the application, which will plot a path from source to destination. The application shows the user's advancement along the route and it issues directions.

2.4 Global System for Mobile communication (GSM)

Global System for Mobile communication is a digital mobile network that is widely used by mobile phone users. GSM uses a variation of time division multiple access and is the most widely used of the three digital wireless telephony technologies: TDMA, GSM and code-division multiple access . GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. In our project we are using it to give alert messages.

2.5 Android Platform

Recently, improvements in smartphone technology have changed the importance of cellular telephones. A phone is not just used for communicating but has also become a crucial part of everyone's daily life. Presently the electronic market is acquired by Android technology. Over time, smartphones and the Android system have become more prevalent. In our project an application is developed to monitor the concentration of air pollutants so that it can be easily monitored by the users. The IoT-Mobair application uses user location data (via GPS system), the Internet of Things (IoT) and sensors to give air quality data. If a user is setting out to a destination, the pollution level of that location is predicted, and a warning is displayed if the pollution level is too high.

CHAPTER 3

METHODOLOGY

3.1 EXISTING SYSTEM

In today's world many pollution monitoring systems are designed by considering different environmental parameters. Existing system model is based wireless sensor networks to monitor physical and environmental conditions with thousands of application in different fields. The sensor nodes directly communicated with the moving nodes deployed on the object of interest which avoided the use of complex routing algorithm but local computations are very minimal. RFID is a means of storing and retrieving data through electromagnetic transmission to an RF compatible integrated circuit. Mobile phones or smart phones those are capable and in built with sensors are applicable for impact on social including how technology of mobile has to be used to protect environmental, sensing and to influence just-in-time information to create movements and actions. Mobile phone sensors were deployed and used on urban areas for monitoring and it was categorized into two major classes, participatory sensing where user is directly involved and opportunistic sensing where user is not involved, but its limitation includes power and static information processing or mobility restrictions. A Wireless Sensor Network inbuilt of many wireless sensors those are inexpensive, which are able to collect, store, process an environmental information, and to communicate with nodes those are neighbors to each other.

3.1.1 Drawback of existing system

The drawbacks of the conventional monitoring instruments are their large size, heavy weight and extraordinary expensiveness. These lead to sparse deployment of the monitoring stations. In order to be effective, the locations of the monitoring stations need careful placement because the air pollution situation in urban areas is highly related to human activities (e.g. construction activities) and location-dependent (e.g., the traffic choke-points have much worse air quality than average). IOT Based Air Pollution Monitoring System monitors the Air Quality over a web server using internet and will trigger an alarm when the air quality goes down beyond a certain level, means when there are amount of harmful gases present in the air like CO2, smoke, alcohol, benzene, NH3, NO and LPG. In our IOT project, it can monitor the pollution level from anywhere using your mobile. This system can be installed anywhere and can also trigger some device when pollution goes beyond some level.

3.2 PROPOSED SYSTEM

IoT-Mobair App for pollution management, industries based specifically, streamline and integrate environmental processes, including air emissions analyses, water and energy management, and waste reduction. Such apps provide visibility for users, specifically industries into the risk of incidents such as chemical leaks, oil spills and toxic substances improper disposal, while compliance strengthening with environmental standards and regulations. Development outsources assistance for such apps that can cover areas like:

- Common platform to track and manage initiative environment across organizations
- Track compliance with ISO 14001 and other industry specific environmental standards
- Real-time environmental processes view, audit incident and findings
- Trigger notifications for threshold breaches
- Risk highlights related with threshold breaks

The app had Air Quality Monitoring following features,

- Indices of air quality for a specific city using real-time computation, air quality daily forecast
- Specify reports for air quality measures based on locations, air quality maps generation.

The design of the air pollution monitoring system involves three main phases:

Phase 1: detect the concentration of air pollutants in the area of interest via sensors.

Phase 2: develop a user-friendly and portable interface – an Android application, which the user can use to know the pollution level in his/her particular area.

Phase 3: predict air quality using the analytical module. The proposed air pollution monitoring system is presented in figure 3.1.

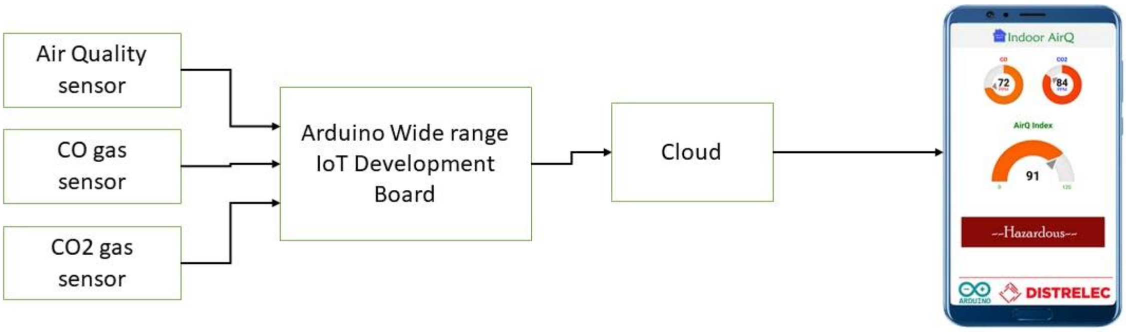

Fig.3.1. System overview

CHAPTER4

SYSTEM DESIGN

4.1 MODULES AND THEIR FUNCTIONALITIES

The design of the air pollution monitoring system involves the following modules:

1. Detect the concentration of air pollutants in the area of interest via sensors.
2. Connecting an Arduino with the help of Node MCU, Wi-Fi module.
3. GSM and GPS Modules
4. Develop a user-friendly and portable interface – an Android application, which the user can use to know the pollution level in his/her particular area.

4.1.1 Detect the concentration of air pollutants in the area of interest via sensors.

The first and major phase of the system is to detect the concentration of air pollutants in the area of interest via sensors. Sensor technology is a supporting technology in sensor networks. For air pollution monitoring, sensors detecting CO_2,CO,NO_x,SO_x, H_2S, NH_3, Cl_2, O_3 and particulate matter are frequently used. Some contaminants can be detected directly by sensors, especially in air environment, such as CO_x, NO_x, and SO_x, but some sensors can only detect a type of pollutants and cannot measure the content of a specific contaminant. For example, the dissolved oxygen (DO) sensor, the pH sensor, and the electrical conductivity sensor fall into this category.

A WSN is an organization of sensor nodes, which gather data using wireless technology. WSNs have limited computation power and memory. WSNs are used to monitor low-frequency data at remote locations. The sensor network layout is intended to recognize all the gases associated with pollution in a given location. Unlike the customary wired framework, expenses of WSN would be minimal. By including more devices, the network can be scaled without much re-work or complex modifications. Moreover, WSN can powerfully adjust to evolving situations.

The project system uses three gas sensors:

1. MQ5Gas Sensor
2. MQ2 gas sensor
3. MQ135 gas sensor

Mq5Gas Sensor: -

Gas Sensor (MQ5) module is useful for gas leakage detection (in home and industry). It is suitable for detecting H2, LPG, CH4, CO, Alcohol. Due to its high sensitivity and fast response time, measurements can be taken as soon as possible. The sensitivity of the sensor can be adjusted by using the potentiometer.

When a gas interacts with this sensor, it is first ionized into its constituents and is then adsorbed by the sensing element. This adsorption creates a potential difference on the element which is conveyed to the processor unit through output pins in form of current. The gas sensor module consists of a steel exoskeleton under which a sensing element is housed. This sensing element is subjected to current through connecting leads. This current is known as heating current through it the gases coming close to the sensing element get ionized and are absorbed by the sensing element. This changes the resistance of the sensing element which alters the value of the current going out of it.

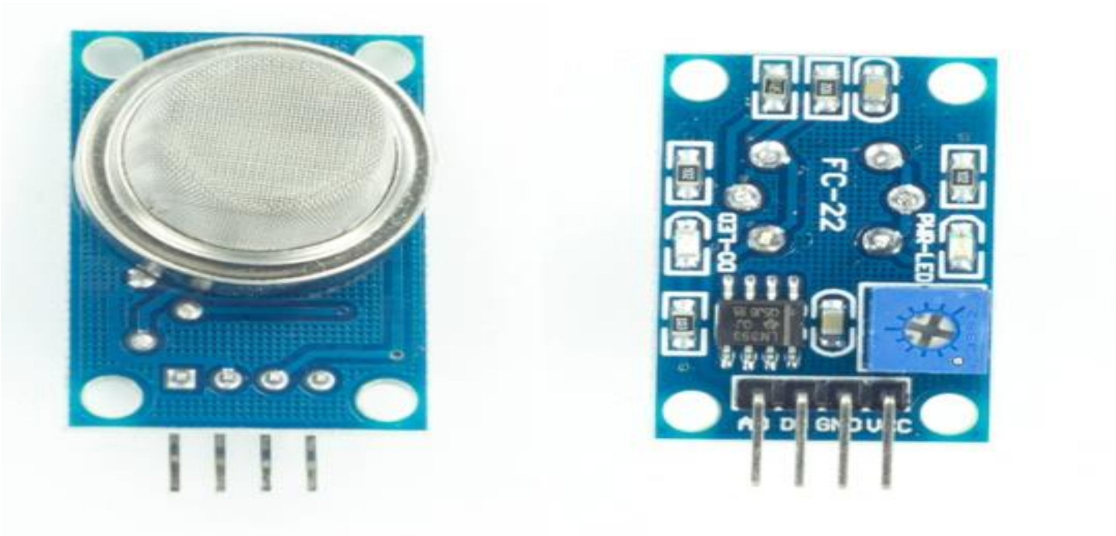

Fig 4.1 MQ5 Gas Sensor Module

Mq2 Gas Sensor: -

MQ2 gas sensor is an electronic sensor used for sensing the concentration of gases in the air such as LPG, propane, methane, hydrogen, alcohol, smoke and carbon monoxide. MQ2 gas sensor is also known as chemiresistor. It contains a sensing material whose resistance changes when it comes in contact with the gas.

The resistance of the sensor is different depending on the type of the gas. The smoke sensor has a built-in potentiometer that allows you to adjust the sensor sensitivity according to how accurate you want to detect gas.

Fig 4.2 MQ2 gas sensor

The voltage that the sensor gives changes accordingly to the smoke/gas level that exists in the atmosphere. The sensor outputs a voltage that is proportional to the concentration of smoke/gas.

In other words, the relationship between voltage and gas concentration is the following:

- The greater the gas concentration, the greater the output voltage.
- The lower the gas concentration, the lower the output voltage.

The output can be an analog signal (A0) that can be read with an analog input of the Arduino or a digital output (D0) that can be read with a digital input of the Arduino.

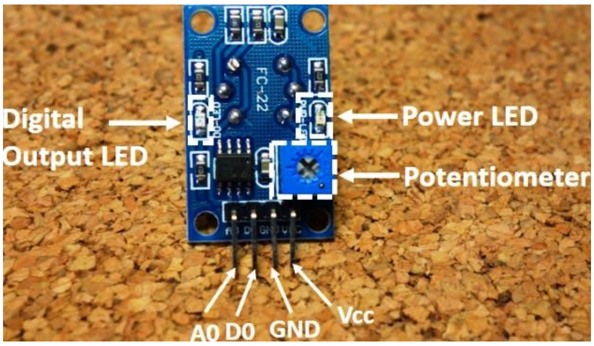

Fig 4.3 Sensor Pin-out details

<u>Mq135 Gas Sensor: -</u>

It is an air quality sensor for detecting a wide range of gases, including NH3, NOx, alcohol, benzene, smoke and CO2. It is ideal for use in office or factory. MQ135 gas sensor has high sensitivity to Ammonia, Sulfide and Benzene steam, also sensitive to smoke and other harmful gases.

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 higher along with the gas concentration rising.

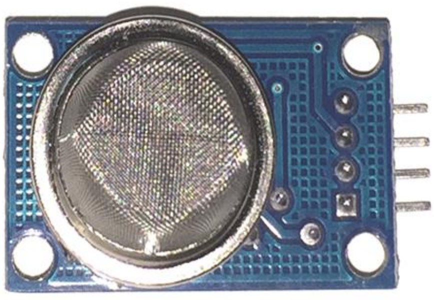

Fig 4.4 MQ-135 Gas Sensor/Module

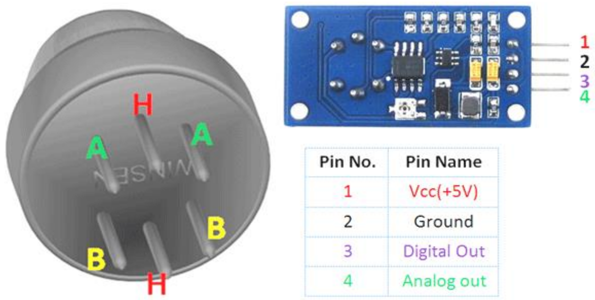

Fig 4.5 MQ-135 Gas Sensor Pin out

4.1.2 Connecting an Arduino with the help of Node MCU, Wi-Fi module.

The Arduino Uno is an open-source microcontroller board based on the microchip atmega328p microcontroller and developed by Arduino Cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.

Node MCU is an open-source firmware and development kit that helps you to prototype or build IoT products. It includes firmware that runs on the ESP8266 Wi-Fi soc, and hardware which is based on the ESP-12 module.

Steps to Setup Arduino IDE for Node MCU Esp8266

- Step 1: Installing Arduino IDE Software
- Step 2: Arduino IDE Icon: After installing Arduino IDE icon is created on the Desktop as shown in the figure.

Fig 4.6 Arduino Icon

- Step 3: Opening Arduino IDE: Click on the Icon to open the Arduino window as shown in the figure.

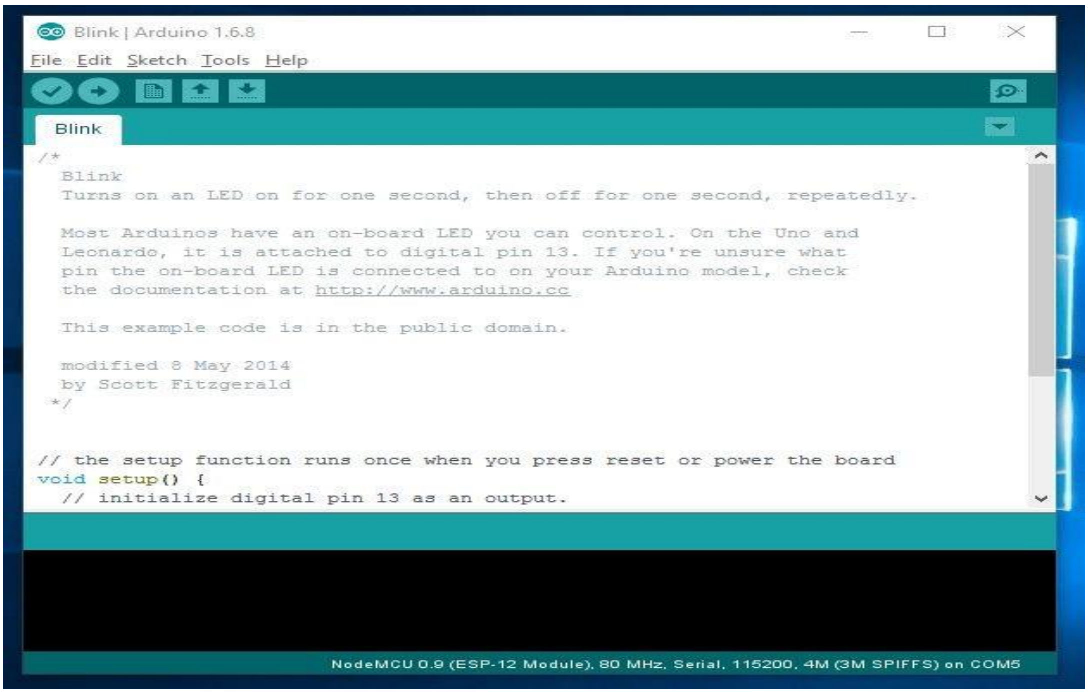

Fig 4.7 Arduino window

- Step 4: Preferences: Open the file and click on the preferences.
- Step 5: Adding ESP8266 Board Manager

In the Additional Boards Manager enter below URL.

http://arduino.esp8266.com/stable/package_esp8266com_index.json

- Step 6: Selecting Board: Now open the tools in that select Board, “Arduino/Genuino Uno” and click on the Boards Manager.
- Step 7: ESP8266 Board Package. The Boards Manager window opens, scroll the window page to bottom till you see the module with the name ESP8266. Once we get it, select that module and select version and click on the Install button. When it is installed it shows Installed in the module as shown in the figure and then closes the window.

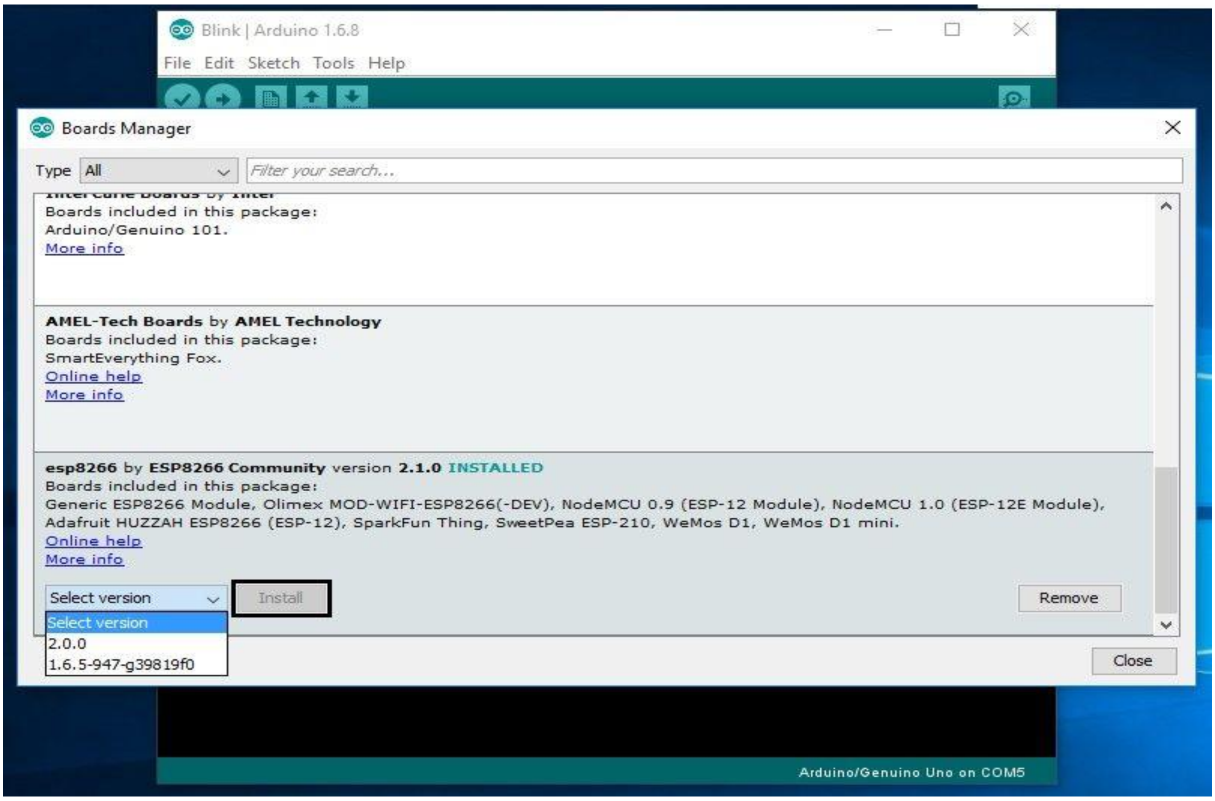

Fig 4.8 Arduino Installation Window

- Step 8: Selecting ESP8266 Arduino Board: To run the esp8266 with Arduino we have to select the Board: “Arduino/Genuino Uno” and then change it to Node MCU 1.0 (ESP-12E Module) or other esp8266 modules depending on what you have. This can be done by scrolling down.
- Step 9: Connecting ESP8266 to the PC: Now let’s connect the ESP8266 module to your computer through USB cable. When module is connected to the USB, COM port is detected.
- Step 10: Selecting Example Program in Arduino IDE.
- Step 11: Selecting COM Port
- Step 12: Uploading the Program to ESP8266 Module
- Step 13: Adding Libraries.

  In case you need to add the libraries to the Arduino follow the example path is shown in the figure C:\Users\Armtronix\Documents\Arduino\libraries. Enter into the libraries folder then paste the file in that as shown.

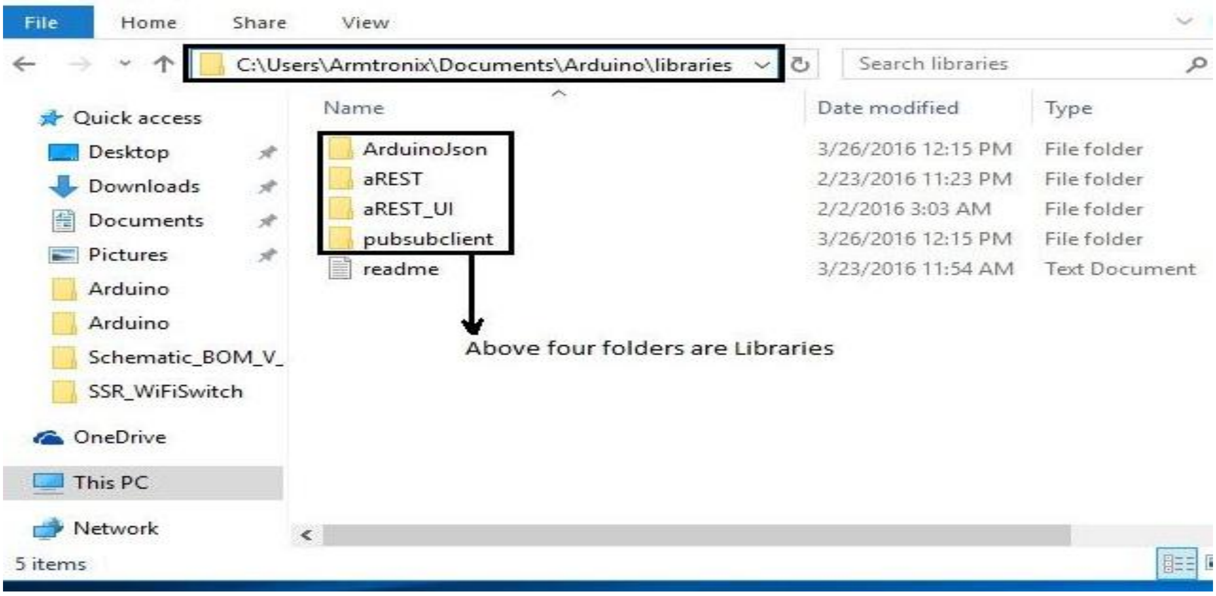

Fig 4.9 Path to add libraries in Arduino

4.1.3 GSM and GPS Modules:

GSM Modules come with AT Command capabilities. Connecting any Microcontroller board to a GSM Module is as simple as connecting power pins and interfacing with UART. To connect to the internet using a GSM module, simply connect the 5V and Gnd from the Node MCU to the GSM Module.

To connect Node MCU to the internet using a GSM module following steps are to be done:

- Connect the 5V and Gnd from the Node MCU to the GSM Module (make sure it works with 5V first)
- Connect the RX of the Node MCU to the TX of the GSM Module
- Connect the TX of the Node MCU to the RX of the GSM Module
- Using Serial.print, send the AT commands to the GSM module.
- Make sure to insert the SIM card properly.

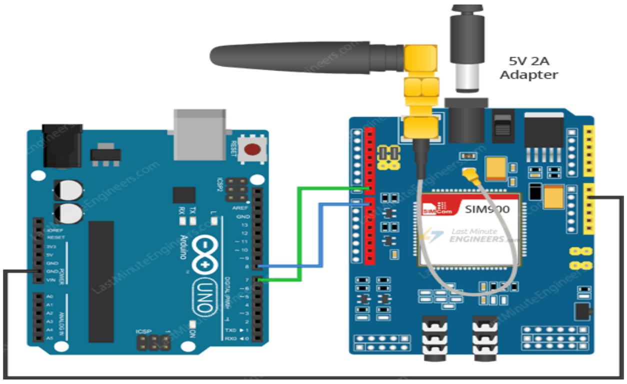

Fig 4.10 GSM module

GPS: Stands for "Global Positioning System." **GPS** is a satellite navigation system used to determine the ground position of an object. **GPS** receivers use a constellation of satellites and ground stations to compute position and time almost anywhere on earth. With this information and some math, a ground based **receiver** or **GPS module** can calculate its position and time.

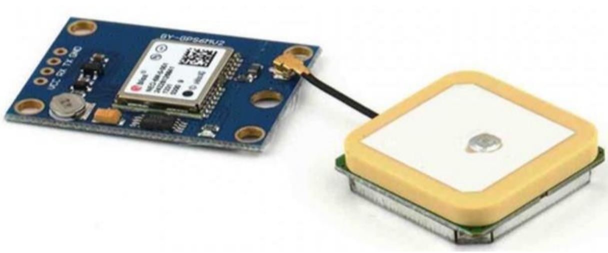

Fig 4.11 GPS

To connect your GPS module to Arduino, use a +5V from the power side of the Arduino and any ground pin. Any two pins will work for the serial communication, but here we are using 3 and 4:

- Connect Arduino pin **3** to the **RX** pin of the GPS Module.
- Connect Arduino pin **4** to the **TX** pin of the GPS Module.

The Overall Connection of Arduino with Node MCU, GPS Module, GSM Module and Sensors (MQ5, MQ2 and MQ-135) Is Shown Below in the Circuit Diagram

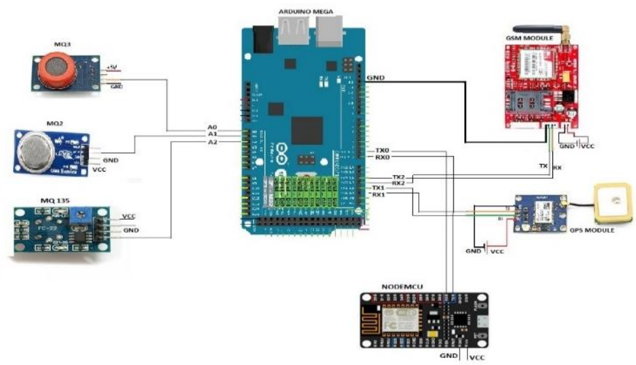

Fig 4.12 Circuit diagram

4.1.4 Develop a user-friendly and portable interface – an Android application, which the user can use to know the pollution level in his/her particular area.

Here we are creating a web application and converting it in to mobile application. An interactive computer program, built with web technologies (HTML, JS), which stores (Database, Files) and manipulates data, and is used by a team or single user to perform tasks over the internet.

Backend language (PHP) - control how your web app works.

Web front end (HTML, Java Script) - for the look and feel of your web app.

Front end of the application is created using Angular java script (framework), Ionic library is used since Ionic is focused mainly on the look and feel, and UI interaction of your app. Capacitor plugins is used to gain access to host operating systems features such as Camera, GPS, Flashlight, etc. Users can build their apps, and they can then be customized for Android, IOS, Windows, Desktop Electron, or modern browsers. Ionic allows app building and deployment by wrapping around the build tool Cordova or Capacitor with a simplified 'ionic' command line tool.

Ionic includes mobile components, typography, interactive paradigms, and an extensible base theme.

Using Web Components, Ionic provides custom components and methods for interacting with them. One such component, virtual scroll, allows users to scroll through a list of thousands of items without any performance hits. Another component, tabs, creates a tabbed interface with support for native-style navigation and history state management.

Besides the SDK, Ionic also provides services that developers can use to enable features, such as code deploys, automated builds. Ionic also provides its own IDE known as Ionic Studio.

Ionic also provides a command-line interface (CLI) to create projects. The CLI also allows developers to add Cordova plugins and additional front-end packages, enable push notifications, generate app Icons and Splash screens, and build native binaries.

Back end is created using PHP frame work, Laravel. Laravel is a server-side PHP framework; it can build full-stack apps, meaning apps with features typically requiring a backend, such as user accounts, exports, order management, etc. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as authentication, routing, sessions, and caching.

CHAPTER 5

IMPLEMENTATION

Internet of Things Mobile–Air Pollution Monitoring System (IoT-Mobair). This kit can be physically placed in various cities to monitoring air pollution. The gas sensors gather data from air and forward the data to the Arduino IDE. The Arduino IDE transmits the data to the server via the Wi-Fi module.

5.1 IMPLEMENTATION

This section depicts the implementation details and tools used in our project work. The proposed system constitutes of three main discrete phases.

Phase 1: detect the concentration of air pollutants in the area of interest via sensors.

Phase 2: develop a user-friendly and portable interface – an Android application, which the user can use to know the pollution level in his/her particular area.

Phase 3: predict air quality using the analytical module.

In the hardware section, three sensors are used to collect data from different area.

1. MQ5Gas Sensor
2. MQ2 gas sensor
3. MQ135 gas sensor

The gases like Carbon Dioxide, Ammonium and Methane will have different normal level of existence, and if this level is exceeded then an alert will be given via the GSM module ie, an alert condition is added to know if the pollution according to these gases are getting high or not. And then by using the GPS module the location can be taken using latitude and longitude values. The gas sensor value/data are uploaded using a Wi-Fi module ESP8266 in the Arduino. That is to the corresponding data base.

The values from the gas sensors are uploaded to the Arduino by the POST method. POST is used to send data to a server to create/update a resource. For example, publish sensor readings to a server.

For the prediction, Machine learning with python software is used and the method used in our project is linear regression. According to the input data set/values along with their percentage, the output value/predicted values will be generated. With GET or POST Method the corresponding URL for integrating with the IoT Mob-air Application will be done. Both GET and POST method is used to transfer data from client to server in HTTP protocol but Main difference between POST and GET method is that GET carries request parameter appended in URL string while POST carries request parameter in message body which makes it more secure way of transferring data from client to server in http protocol.

5.1.1 POST method

The POST request method requests that a web server accepts the data enclosed in the body of the request message, most likely for storing it. It is often used when uploading a file or when submitting a completed web form.

5.1.2 GET method

GET method is used to appends form data to the URL in name or value pair. By using GET, the length of URL will remain limited. It helps users to submit the bookmark the result. GET is better for the data which does not require any security. The HTTP GET request method retrieves information from the server.

5.1.3 Machine learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

5.1.4 Linear regression

Linear regression is a statistical approach for modeling relationship between a dependent variable with a given set of independent variables. Linear regression models are used to show or predict the relationship between two variables or factors. The factor that is being predicted is called the dependent variable.

5.2 ALGORITHM

Training

1. Import the Libraries
2. Read the CSV file (Dataset)
3. Train the data
4. Classify the data
5. Plot the accuracy
6. Create the Model

Test

1. Import the Libraries
2. Define the classifier
3. Read the CSV file
4. Predict the output

5.3 HARDWARE REQUIREMENTS

- Arduino Uno
- Node MCU
- MQ5Gas Sensor
- MQ2 gas sensor
- MQ135 gas sensor
- Global Positioning System (GPS)
- Global System for Mobile communication (GSM)
- Android Device
- Server

5.4 SOFTWARE REQUIREMENTS

- Programming Language (Python).
- Machine Learning algorithm
- Arduino IDE
- Node JS
- Framework: Angular, Ionic (app frontend)

- Capacitor
- PHP
- Framework: Laravel (app backend)
- Visual Studio IDE
- HTML- Kit
- Composer
- XAMPP

5.5 CODE

5.5.1 Sample Arduino code

```
#include <TinyGPS++.h>
TinyGPSPlusgps;
#include <dht.h>
dht DHT;
int dhtpin=2;
#define gas1 A0
#define gas2 A1
#define gas3 A2
volatile float minutes, seconds;
volatile int degree, secs, mins;
String messageString;
int temp, humi;
int val1,val2,val3;
String str;
void setup () {
Serial.begin(9600);
  Serial1.begin(9600);
  Serial2.begin(9600);
pinMode(dhtpin,INPUT);
pinMode(gas1,INPUT);
pinMode(gas2,INPUT);
pinMode(gas3,INPUT);
```

```
  // put your setup code here, to run once:

}

void loop () {

smartDelay (1000);
  unsigned long start;
  double lat_val, lng_val, alt_m_val;
  bool loc_valid, alt_valid;
lat_val = gps.location.lat ();
loc_valid = gps.location.isValid();
lng_val = gps.location.lng ();
alt_m_val = gps.altitude.meters();
alt_valid = gps.altitude.isValid();
DegMinSec(lat_val);
DegMinSec(lng_val);
  String latt = String(lat_val,6);
  String lngg = String(lng_val,6);
messageString = "http://www.google.com/maps/place/"+latt+","+lngg;
  //Serial.println(messageString);
Serial.println(messageString);



  DHT.read11(dhtpin);
humi= DHT.humidity;
 temp= DHT.temperature;
// Serial.print("Tempeature=");
// Serial.println(temp);
// Serial.print("Humidity=");
// Serial.println(humi);

  val1 = analogRead(gas1);
  val2 = analogRead(gas2);
```

```
  val3 = analogRead(gas3);
  String data1= String(val1);
if (data1.length() <3)
  {
    data1="0"+data1;
  }
   String data2= String(val2);
if (data2.length() <3)
  {
    data2="0"+data2;
  }
   String data3= String(val3);
if (data3.length() <3)
  {
    data3="0"+data3;
  }

//   Serial.println(data1);
//   Serial.println(data2);
//   Serial.println(data3);
// Serial.println(val1);
// Serial.println(val2);
// Serial.println(val3);
  //delay (1000);
  str        =String(humi)+","+String(temp)+","+String(data1)        +","+String(data2)
+","+String(data3);
Serial.println(str);

  if (val1> 100)
  {
Serial.println("message");
   Serial2.println("AT");
delay (100);
   Serial2.println("AT+CMGF=1") ;//Sets the GSM Module in Text Mode
```

```
delay (1000) ;// Delay of 1000 milli seconds or 1 second
  Serial2.println("AT+CMGS=\"+91xxxxxxxxxx\"\r"); // Replace x with mobile number
delay (2000);
  Serial2.println(messageString); // The SMS text you want to send
delay (1000);
 Serial2.println((char)26) ;// ASCII code of CTRL+Z
delay (100);
 }
 if (val2> 100)
 {
Serial.println("message");
   Serial2.println("AT");
delay (100);
  Serial2.println("AT+CMGF=1");  //Sets the GSM Module in Text Mode
delay(1000);                    // Delay of 1000 milli seconds or 1 second
  Serial2.println("AT+CMGS=\"+91xxxxxxxxxx\"\r"); // Replace x with mobile number
delay(2000);
  Serial2.println(messageString);        // The SMS text you want to send
delay(1000);
 Serial2.println((char)26);            // ASCII code of CTRL+Z
delay(100);;
 }
 if (val3> 100)
 {
Serial.println("message");
   Serial2.println("AT");
delay(100);
  Serial2.println("AT+CMGF=1");            //Sets the GSM Module in Text Mode
delay(1000);                    // Delay of 1000 milli seconds or 1 second
  Serial2.println("AT+CMGS=\"+91xxxxxxxxxx\"\r"); // Replace x with mobile number
delay(2000);
```

```
  Serial2.println(messageString);            // The SMS text you want to send
delay(1000);
 Serial2.println((char)26);                  // ASCII code of CTRL+Z
delay(100);
  }

}

static void smartDelay(unsigned long ms)
{
  unsigned long start = millis();
  do
  {
    while (Serial1.available())
gps.encode(Serial1.read());
  } while (millis() - start <ms);
}

void DegMinSec( doubletot_val)
{
  degree = (int)tot_val;
  minutes = tot_val - degree;
  seconds = 60 * minutes;
  minutes = (int)seconds;
  mins = (int)minutes;
  seconds = seconds - minutes;
  seconds = 60 * seconds;
  secs = (int)seconds;
}
```

5.5.2 Python Program using linear regression

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
dataset = pd.read_csv('pollution.csv')

#Reading values for conc of NH4

X1 = dataset.iloc[ : ,   1: 2 ].values
Y1 = dataset.iloc[ : ,  2:3 ].values


#Reading values for conc of CO

X2 = dataset.iloc[ : ,   3: 4 ].values
Y2 = dataset.iloc[ : ,  4:5 ].values


#Reading values for conc of CH4

X3 = dataset.iloc[ : ,   5: 6 ].values
Y3 = dataset.iloc[ : ,  6:7 ].values


from sklearn.model_selection import train_test_split

#Splitting Values of mode 1
X1_train, X1_test, Y1_train, Y1_test = train_test_split( X1, Y1,
test_size = 1/5, random_state = 0)

#Splitting Values of mode 2
X2_train, X2_test, Y2_train, Y2_test = train_test_split( X2, Y2,
test_size = 1/5, random_state = 0)


#Splitting Values of mode 3
X3_train, X3_test, Y3_train, Y3_test = train_test_split( X3, Y3,
test_size = 1/5, random_state = 0)


from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

#trained model 1
regressor1 = regressor.fit(X1_train, Y1_train)
#trained model 2
regressor2 = regressor.fit(X2_train, Y2_train)
#trained model 3
regressor3 = regressor.fit(X3_train, Y3_train)




mode=input("Please enter the gas, to check pollution \n"
           "1 : NH4 , 2 : CO , 3 : CH4   Enter :: ")

if mode=="1":
    NH4=input("Please enter the NH4 gas value, to check :: ")
    Y1_pred = regressor1.predict([[int(NH4)]])
print("Pollution predicted NH4 : ",Y1_pred)
```

```
if mode=="2":
    CO=input("Please enter the CO gas value, to check :: ")
    Y1_pred = regressor2.predict([[int(CO)]])
print("Pollution predicted CO : ",Y1_pred)

if mode=="3":
    CH4=input("Please enter the CH4 gas value, to check :: ")
    Y1_pred = regressor3.predict([[int(CH4)]])
print("Pollution predicted CH4 : ",-Y1_pred)
```

CHAPTER 6

RESULT AND ANALYSIS

Fig 6.1 Home Page

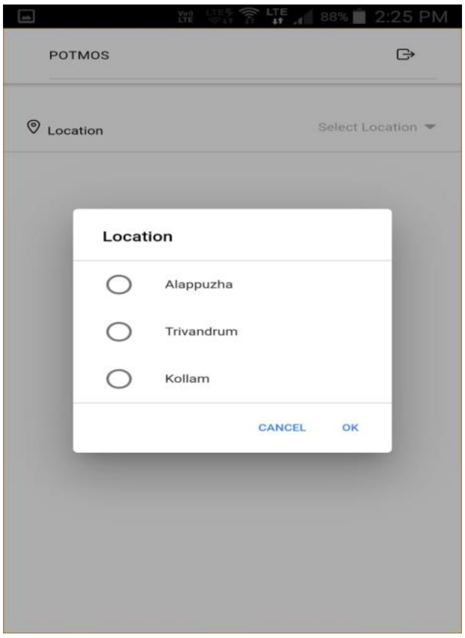

Fig 6.2 Location selecting

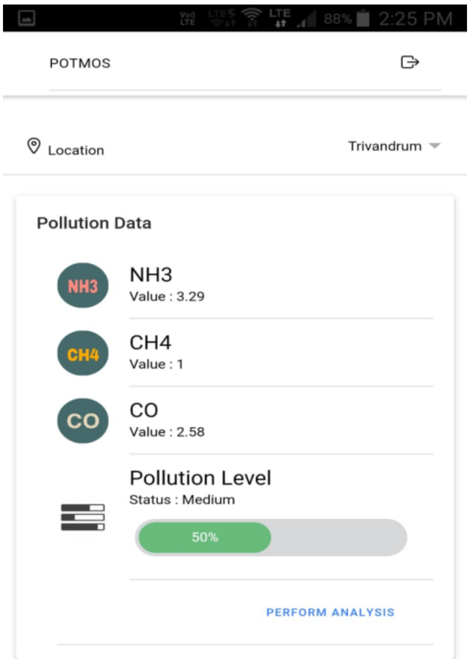

Fig 6.3 Pollution data

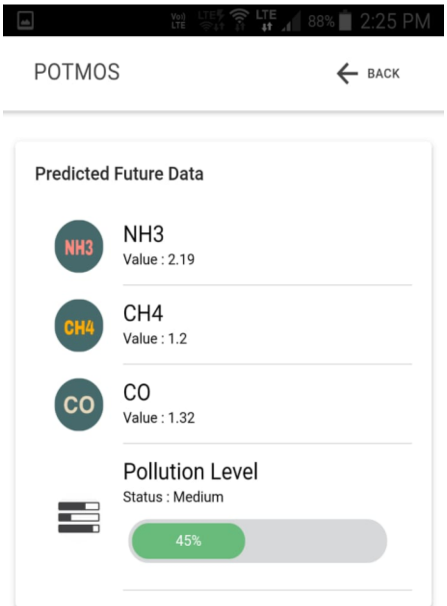

Fig 6.4 Predicted future data

CHAPTER 7
ADVANTAGES AND APPLICATIONS

7.1 ADVANTAGES

- The data collected from air quality monitoring helps us assess impacts caused by poor air quality on public health.
- Quality of air in an area can be checked indoors as well as outdoors
- Mobile air quality monitoring techniques are allowing us to intensively describe air quality, rather than relying on fixed sites to provide a representative picture of air quality.
- Mobile monitoring allows more opportunity to assess real time air quality in an area; mobile application can be moved around in multiple locations.
- Can cover more location quickly in an easier and cheaper way.
- Simple, compact and easy to handle

7.2 APPLICATIONS

- To make this data of pollutants available to common people.
- For knowing roadside pollution and pollution level of a desired location or area.
- Industrial pollutants level monitoring.
- To predict future pollution range.

CHAPTER 8
CONCLUSION

Pollution in earlier days was negligible. Currently, however, pollution is increasing day-by-day because of various reasons such as industrial growth, development of automobile industries, and chemical industries. Therefore, to protect humans from harmful gasses, the air pollution monitoring system is developed that helps a person to detect, monitor, and test air pollution in a given area.

An IoT kit was prepared using gas sensors, Arduino IDE (Integrated Development Environment), and a Wi-Fi module. The gas sensors gather data from air and forward the data to the Arduino IDE. The Arduino IDE transmits the data to the cloud via the Wi-Fi module. An Android application can access relevant air quality data from the cloud. If a user is traveling to a destination, the pollution level of that destination is predicted, and a warning is displayed if the pollution level is too high. Air quality data can be used to predict future air quality index (AQI) levels.

REFERENCES

[1] Internet of Things Mobile - Air Pollution Monitoring System, Swati Dhingra, Rajasekhara Babu Madda, Amir H. Gandomi, Senior Member, IEEE, Rizwan Patan, Mahmoud Daneshmand, Life Member, IEEE

[2] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, “Field calibration of a cluster of low-cost commercially available sensors for air quality monitoring. Part B: NO, CO and CO2,” Sensors Actuators B Chem., vol. 238, pp. 706–715, 2017.

[3] S. Gaglio, G. Lo Re, G. Martorella, D. Peri, and S. D. Vassallo, “Development of an IoT environmental monitoring application with a novel middleware for resource constrained devices,” in Proceedings of the 2nd Conference on Mobile and Information Technologies in Medicine (MobileMed 2014), 2014.

[4] G. Lo Re, D. Peri, and S. D. Vassallo, “Urban air quality monitoring using vehicular sensor networks,” in Advances onto the Internet of Things, Springer, 2014, pp. 311–323.

[5] R. Peterová and J. Hybler, “Do-it-yourself environmental sensing,” Procedia Comput. Sci., vol. 7, pp. 303–304, 2011.

[6] M. R. B. and A. J. B. Alok N. Bhatt, “Automation Testing Software that Aid in Efficiency Increase of Regression Process,” Recent Patents Comput. Sci., vol. 6, no. 2, pp. 107–114, 2013.

[7] B. Predić, Z. Yan, J. Eberle, D. Stojanovic, and K. Aberer, “Exposure sense: Integrating daily activities with air quality using mobile participatory sensing,” in Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on, 2013, pp. 303–305.

[8] Y. Zheng, X. Chen, Q. Jin, Y. Chen, X. Qu, X. Liu, E. Chang, W.-Y. Ma, Y. Rui, and W. Sun, “A cloud-based knowledge discovery system for monitoring fine-grained air quality,” Prep. Microsoft Tech Report, http//research. Microsoft. com/apps/pubs/default. aspx, 2014.

[9] G. Lo Re, D. Peri, and S. D. Vassallo, “A mobile application for assessment of air pollution exposure,” in Proceedings of the 1st Conference on Mobile and Information Technologies in Medicine (MobileMed 2013), 2013.

[10] A. Tamayo, C. Granell, and J. Huerta, “Using SWE standards for ubiquitous environmental sensing: a performance analysis,” Sensors, vol. 12, no. 9, pp. 12026–12051, 2012.

[11] A. A. Reshi, S. Shafi, and A. Kumaravel, “VehNode: Wireless Sensor Network platform for automobile pollution control,” in Information & Communication Technologies (ICT), 2013 IEEE Conference on, 2013, pp. 963–966.

[12] T. H. Mujawar, V. D. Bachuwar, and S. S. Suryavanshi, “Air Pollution Monitoring System in Solapur City using Wireless Sensor Network,” Proc. Publ. by Int. J. Comput. Appl. CCSN-2013, pp. 11–15, 2013.

[13] A. De Nazelle, E. Seto, D. Donaire-Gonzalez, M. Mendez, J. Matamala, M. J. Nieuwenhuijsen, and M. Jerrett, “Improving estimates of air pollution exposure through ubiquitous sensing technologies,” Environ. Pollut., vol. 176, pp. 92–99, 2013.

[14] a Zanella, N. Bui, a Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for Smart Cities,” IEEE Internet Things J., vol. 1, no. 1, pp. 22–32, 2014.

[15] J. A. Stankovic, “Research directions for the internet of things,” IEEE Internet Things J., vol. 1, no. 1, pp. 3–9, 2014.

[16] D. Uckelmann, M. Harrison, and F. Michahelles, “An architectural approach towards the future internet of things,” in Architecting the internet of things, Springer, 2011, pp. 1–24.

[17] M. Schmidt, Arduino: a quick-start guide. Pragmatic Bookshelf, 2015.