

DEPARTMENT: BIO MEDICAL ENGINEERING

YEAR: THIRD YEAR

PROJECT SUBMISSION PHASE -3

TOPIC – AIR QUALITY MONITORING

Team members

1.KALIDASS.P

2.KUMARAGURU.M

3.LAWRANCE.N

4.MAHESH.R

5.MANOJKUMAR.S

6.MARIARAJ.G

By.

MANOJKUMAR S

AIR QUALITY MONITORING

Introduction:

Air quality monitoring refers to continuous measurement of specific air pollutants also known as “criteria air pollutants”. Obtained air pollution data together with natural background/trace gas monitoring and stationary source emission monitoring helps to define what kind of air pollution people are exposed to.

IoT enabled proactive indoor air quality monitoring system for sustainable health

management:-

This paper proposes an IoT based indoor air quality monitoring system for tracking the ozone concentrations near a photocopy machine. The experimental system with a semiconductor sensor capable of monitoring ozone concentrations was installed near a high volume photocopier. The IoT device has been programmed to collect and transmit data at an interval of five minutes over blue tooth connection to a gateway node that in turn communicates with the processing node via the Wi-Fi local area network.

HARDWARE REQUIREMENTS:

 For Different Parameter Sensing:-

• Temperature and Humidity sensor (DHT11)

• Air Quality sensor (MQ 135)

• 2n2222 Transistor

• DC Fan

• Potentiometer

• 16x2 LCD Panel

• NodeMCU

• Arduino Uno

 For Power Supply:-

• Step down transformer (12-0-12 V,1 A)

• Diodes

• Voltage Regulator (7805)

• Capacitors (0.01 micro Farad, 470 micro Farad)

• Wires

SOFTWARE REQUIREMENTS:

• Arduino (Version 1.8.2)

• THINGSPEAK website

OBJECTIVE:

 To measure and display temperature and humidity level of the environment.

 To combine advanced detection technologies to produce an air quality sensing system with advanced capabilities to provide low cost comprehensive monitoring.

 To display the sensed data in user friendly format in LCD display panel.

PYTHON CODE FOR AN AIR QUALITY MONITORING SYSTEM THAT USES MACHINE LEARNING TO PREDICT AIR QUALITY LEVELS

# Import necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.model\_selection import train\_test\_split

# Load the air quality dataset

df = pd.read\_csv('air\_quality\_data.csv')

# Explore the data

print(df.head())

# Split the data into training and testing sets

X = df[['Temperature', 'Humidity', 'Wind Speed']]

y = df['Air Quality']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train a linear regression model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Evaluate the model on the test set

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mse)

r2 = r2\_score(y\_test, y\_pred)

print("Mean Squared Error: ", mse)

print("Root Mean Squared Error: ", rmse)

print("R-squared: ", r2)

# Visualize the predicted vs actual values

plt.scatter(y\_test, y\_pred)

plt.xlabel('Actual Air Quality')

plt.ylabel('Predicted Air Quality')

plt.title('Air Quality Prediction')

plt.show()