Monitoring, Alerting and Forecasting Vehicle Emissions Using IoT

Manas P
Department of Computer Applications
Amal Jyothi College of Engineering Kanjirappally, India
manasp@mca.ajce.in

Ajith G S
Asst.professor Department of ComputerApplication
Amal Jyothi College of EngineeringKottayam, India
gsajith@amaljyothi.ac.in

***Abstract*: This paper is delivered to test the variations in toxic additives like Carbon Monoxide (CO) inside the atmosphere of Kerala, India. Vehicles are the major agents that produce air pollutants in cities. The Road and Transportation Office (RTO) and Central Pollution Control Board (CPCB) monitor all the activities of the vehicle. So, the primary motive of this paper is to measure the toxicity which is emitted via the exhaust of the vehicles and send the document to RTO, CPCB, and the consumer of the vehicle if it exceeds the pollution limit via Global System for Mobile Communication (GSM) and internet of things (IoT). There is a fitness certificate furnished by the Indian government for every vehicle, so we set the pollution limit on the sensors according to the standards. Thingspeak is used to collect data and IFTTT Applet is used to send alert to user and we forecast the CO emission exceed rate using the dataset got from Thingspeak.**

Keywords: *IoT, Vehicle Emission, GSM, ARIMA, SARIMA*

I. INTRODUCTION

The significant degree of air-contamination in metropolitan regions, caused in no little degree by street transport, requires the execution of persistent and precise observing strategies if outflows are to be limited. Toxic gas emissions account for one-fifth of overall emissions. The sole main cause in India that is contributed by vehicle travel is CO in the atmosphere. To reduce pollution in cities, it is necessary to monitor toxic emissions produced by vehicles and apply short and long-term mitigation strategies.

Unfavorable wellbeing impacts, including asthma, eye aggravation, lung problems and results of richness are the intense and persistent results of traffic related air-contamination. Individuals who live in these spots have a higher danger of contracting diseases.

Currently, the government has established new emission standards for monitoring air pollution, and the data obtained allows for the reduction of the harmful impacts on the environment. The new air quality standards in India to safeguard the society are tabulated in Table 1.

Taxi, transports, and trucks are liable for 72% of CO discharge in the metro urban areas. Because of these disturbing conditions CPCB made FC reestablishment required each year for Heavy Transport Vehicles (HTVs) and five years for Light Motor Vehicles (LMVs). Every three months, every vehicle must be assessed to receive a Pollution Under Control (PUC) certificate, according to the legislation. By monitoring the Air Quality Index using specific sensors we can measure the air pollution. The from sensor is transmitted using the new technologies like IoT and Wireless Sensor Network which will result in quick and reliable notifications. The most concerning scenarios arise when vehicle emissions surpass the standard limits, which may be discovered by improving the method of measuring individual vehicle output. It is critical to have precise mobile and fixed sensing equipment for the process of monitoring the Air Quality Index, which the metro corporation will use to make emission rules more rigorous in order to lower them.

Meanwhile, the usage of e-vehicles in many European nations, particularly Norway and Austria, analyses and comprehends the elements impacting e-vehicle competitiveness and socioeconomic issues. They are also imposing emission regulations as a severe condition and undertaking public awareness efforts. Even while the revenue impacts of e-vehicles are negligible in the long term, the expense of bringing a new technology to market is considerable. In contrast, emerging countries such as India, Brazil, and South Africa rely heavily on fossil fuels for transportation and home use. When a new car is acquired in these nations, an emission certificate is supplied, but most individuals disregard its renewal. This study focuses on the creation of field instruments as a method for increasing attention in all individuals. The prototype provided in this work, which consists of a microcontroller and a sensor to analyze car emissions, communicate through GSM, and inform the government to track the AQI, is exhibited.

Fuel type	**Pollutant gases**	**BS6**	**BS4**
Petroleum distillated vehicle	Nitrogen Oxide (NOx) Limit	60 mg	80 mg
	Particulate Matter (PM) Limit	4.5 mg/km	-
	Carbon Monoxide (CO) Limit	1.0 g/km	1.0 g/km
Diesel fuel vehicle	Nitrogen Oxide (NOx) Limit	80 mg	250 mg
	Particulate Matter (PM) Limit	4.5 mg/km	25 mg
	Carbon Monoxide (CO) Limit	0.05 g/km	0.05 g/km
	HC + NOx	170 mg/km	300 mg

Table I
Emission standards in India

II. RELATED WORKS

Even if CO2 emissions are ceased for 1000 years, the increase in CO2 levels is mostly permanent. Among these permanent

DOI:10.5281/zenodo.6365072
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

effects is a decrease in rainfall, which, combined with high heat waves, has resulted in the establishment of "dust bowl" conditions in various places over the century. The dramatic increase in CO2 concentration led the atmospheric temperature to rise by up to 3.2 degrees Celsius, resulting in thermal expansion of the seas and a rise in sea level of 0.4 to 1.0 meters. As a result of these difficult circumstances, various locations throughout the world will have very little precipitation, as well as significant heat waves that will result in the creation of 'dust bowl' conditions.

A vehicle monitoring system based on on-board IoT was successfully built to collect data from two distinct driving patterns, especially deceleration and speed based on road slope. When compared to the other car, the CO2 emission was consistently lower in the vehicle whose engine oil was changed often. The rate at which CO2 emissions increase and drop is determined by the vehicle's speed. As a result, there is a strong chance of recognizing CO2 emissions based on the interval of changing the engine oil and the frequency are considerably supplied. To react to natural change, the United Nations Framework Convention on Climate Change (UNFCCC) conducts the worldwide most elevated highlight convince creating and created nations to decrease their emanations by at minimum 5% to keep worldwide ecological change under 2 degrees Celsius and stay away from genuine worldwide ecological change before long. Accordingly, the Philippines declared in their Intended Nationally Determined Contributions (INDC) in 2015 that they wanted to decrease their all-out discharges by 2030 by lessening the utilization of oil subordinates for energy, transportation, industry, other native prerequisites. In India, as shown in Fig. 1, over 77 percent of emissions originate from only three sectors: energy, industry, and transportation, which is a massive amount contributing 1.14 tons per capita. These emissions are based on the most recent Social Accounting Matrix available (SAM). Because of uneven and sloppy roads, different types of road patterns have a significant impact on CO2 emissions. The influence of road slope on gasoline consumption has been analyzed, and the fuel economy has been compared. As a result, the emission levels are typically 10% higher on raised steep roads and 2% higher on flat roads.

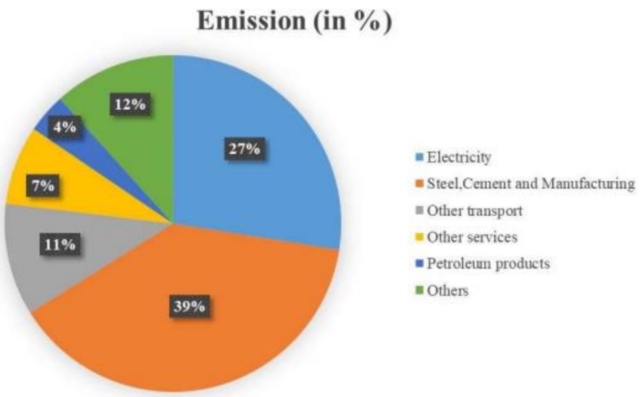

Fig 1. CO2 emission by sectors

Meanwhile, several driving patterns are used to compare fuel consumption and emissions, such as fast acceleration, quick breaking, and driving the vehicle in idle mode, all of which have a significant influence on emissions. It is believed that manual automobiles may save up to 19% of their fuel, while automated vehicles can save up to 7%. The transportation sector, as described in [1], [2], [3], and [4], was the main source of air pollution in various nations. To address this issue, most nations have established Wireless Sensor Networks (WSNs) to identify the most polluted roads and places and to determine the most effective strategies to minimize suspended particulate matter and hazardous gas emissions, which are generating major health issues. Khedo et al employ a WSN with a hierarchical routing protocol to gather the numerous motes data and detect duplicated, distorted, and noisy data, which is then aggregated to generate simple data for processing and determining the air quality. This protocol also assists the motes in sleeping while they are idle.

III. PROPOSED SYSTEM

From previous conversations, it is evident that transportation is one of the biggest contributors to climate change and its numerous negative consequences for all living species. The majority of existing systems use WSN to collect data from motes that have high range noise, delay, and duplicated signals. As a result of the processing and obtaining real-time data, this is a time-consuming operation. These approaches are only useful for determining average air quality rather than identifying particular polluting cars. Existing approaches use additional protocol hierarchy algorithms to filter out noise and duplicated signals, resulting in a more complicated nature in terms of deployment and expense. To avoid this, this study tried to simplify the complexity of obtaining data and offered a low-cost monitoring arrangement.

The suggested system is centered on detecting and monitoring individual vehicle emission levels, as well as notifying the user if the vehicle exceeds the regulatory limit (See Table I). This approach employs a gas sensor capable of detecting the mixture of CO which is mounted on the vehicle's exhaust. The sensor is additionally shielded from heat dissipation by encasing it in the GALV Thermal Isolation Clip, which prevents the sensor from malfunctioning. The technology is GPS-enabled to notify the position to officials in order for them to locate the contaminated car. The prototype is powered by two 6V batteries, which power the whole system, including GSM, Global Positioning System (GPS) module, and indication, with the exception of the gas sensor, which utilizes a single 9V battery independently owing to its high-power consumption. The acquired data is saved in real-time in a Thingspeak and assessed using embedded C code. Any deviation from the usual is conveyed to the driver.

DOI:10.5281/zenodo.6365072
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

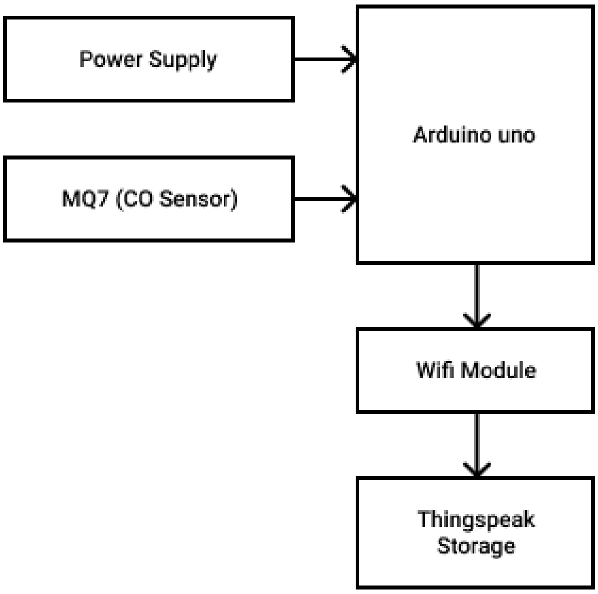

Fig 2. Block diagram of proposed system

The block diagram of the proposed system is shown in Fig.2.

IV. IMPLEMENTATION

Virtually Designed the board using circuito.io using the hardwares: MQ-7 Carbon Monoxide Sensor, QuadBand GPRS Module, BreadBoard, ESP8266-01 Wifi Module, Arduino Uno.

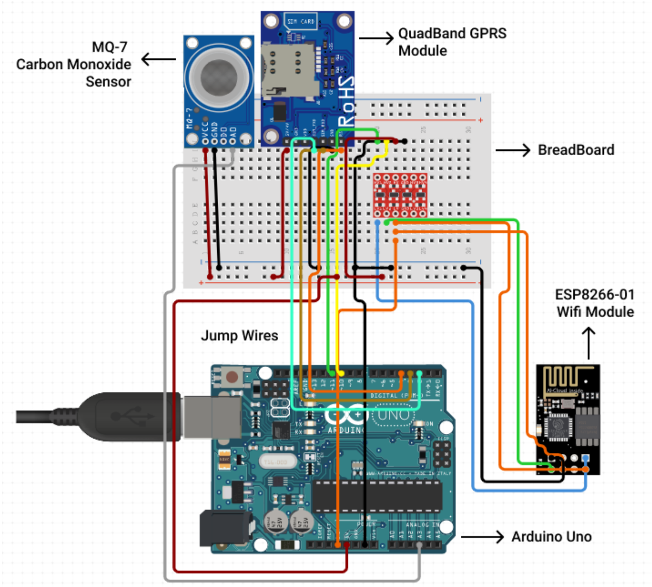

Fig 3. Prototype of the System

Arduino Uno: The Arduino UNO is a low-cost, versatile, and simple-to-use programmable open-source microcontroller board that can be integrated into a wide range of electrical applications. As an output, this board can operate relays, LEDs, servos, and motors and may be interfaced with other Arduino boards, Arduino shields, and Raspberry Pi boards.

MQ-7 Sensor: CO levels in the air are measured. It detects concentrations ranging from 20 to 2000 parts per million.

ESP8266-01: The ESP8266 is a low-cost Wi-Fi board that can be quickly connected to a microcontroller and used to connect any project to the internet (of things).

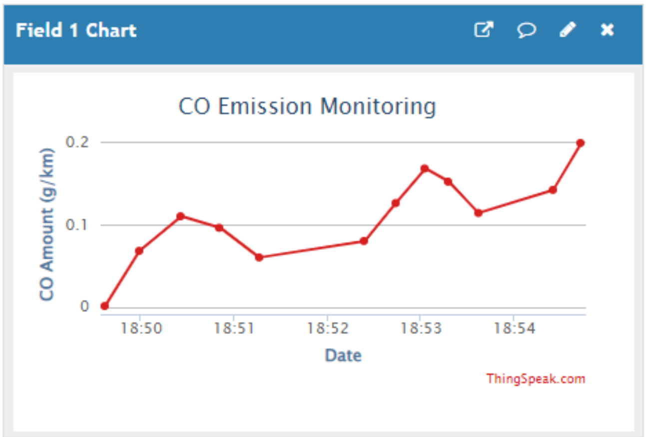

Fig 4. Thingspeak Cloud Data

This graph depicts the emission of Carbon monoxide from the vehicle date wise.
X – Axis: Date
Y – Axis: CO Value

Code to send data from sensor to Thingspeak cloud:

```
#include <stdlib.h>
#include <Arduino.h>
#include <ESP8266.h>

#define CO_CONTENT_INPUT          A0
#define HEATER_5V                 60000
#define HEATER_1_4_V              90000
#define READING_DELAY_MILLIS      5000
```

Fig 5. Define Modules

```
String apiKey="45ULHZHSRO8KK0BI";  //API key of thingspeak
const char *SSID     = "ManasWifi"; // Wi-Fi name
const char *PASSWORD = "manas@123" ; // Wi-Fi password
const char *server = "api.thingspeak.com";
const char *host = "184.106.153.149";
int relayInput = 2;
int buzzerPin = 5;

unsigned long g_startMillis;
unsigned long g_switchTimeMillis;
boolean       g_heaterInHighPhase;
```

Fig 6. Connecting wifi and Thingspeak

DOI:10.5281/zenodo.6365072
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

```
void setup(void){
  Serial.begin(115200);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(SSID);

  WiFi.begin(SSID, PASSWORD);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  pinMode(SENSOR_LED, OUTPUT);

  g_startMillis = millis();

  switchHeaterHigh();

  Serial.println("Elapsed Time (s), Gas Level");
}
```

Fig 7. Setup

Initialize the Serial Communication with baud 115200, connect to the Router with SSID and PASSWORD and turn on the heater on the CO sensor.

```
void loop(void){

  if(g_heaterInHighPhase){
    // 5v phase of cycle. see if need to switch low yet
    if(millis() > g_switchTimeMillis) {
      switchHeaterLow();
    }
  }
  else {
    // 1.4v phase of cycle. see if need to switch high yet
    if(millis() > g_switchTimeMillis) {
      switchHeaterHigh();
    }
  }

  presentGasLevel();
  delay(READING_DELAY_MILLIS);
}
```

Fig 8. Switching Heat Cycle

Switch between the Heating Cycle from 5v to 1.4v and read the Gas Level

```
//Turn on the Heater by Providing 5V to the CO Sensor
void switchHeaterHigh(void){
  // 5v phase
  digitalWrite(SENSOR_LED, LOW);
  g_heaterInHighPhase = true;
  g_switchTimeMillis = millis() + HEATER_5V;
}


//Turn off the Heater by Providing 1.4V to the CO Sensor
void switchHeaterLow(void){
  // 1.4v phase
  digitalWrite(SENSOR_LED, HIGH);
  g_heaterInHighPhase = false;
  g_switchTimeMillis = millis() + HEATER_1_4_V;
}
```

Fig 9. Turning on and off heater

```
void presentGasLevel(){
  unsigned int l_gasLevel = analogRead(CO_CONTENT_INPUT);
  unsigned int time = (millis() - g_startMillis) / 1000;
  //Serial.print(l_gasLevel);
  WiFiClient client;
  const int l_httpPort = 80;
  if (!client.connect(g_host, l_httpPort)) {
    Serial.println("Thingspeak Connection Failed");
    return;
  }

  String url = "/update?key=";
  url += g_apiKey;
  url += "&field1=";
  url += String(l_gasLevel);
  //Serial.println(url);

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  delay(10);

  if(l_gasLevel>1){
    digitalWrite(relayInput, HIGH);
    delay(1000);
    digitalWrite(buzzerPn, HIGH);
  }
  if(l_gasLevel<1){
    digitalWrite(relayInput, LOW);
    delay(1000);
    digitalWrite(buzzerPn, LOW);
  }

  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }
  Serial.println("ThingSpeak Connection Closed");
  delay(10000);
}
```

Fig 10. Reading and updating data on Thingspeak

DOI:10.5281/zenodo.6365072
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

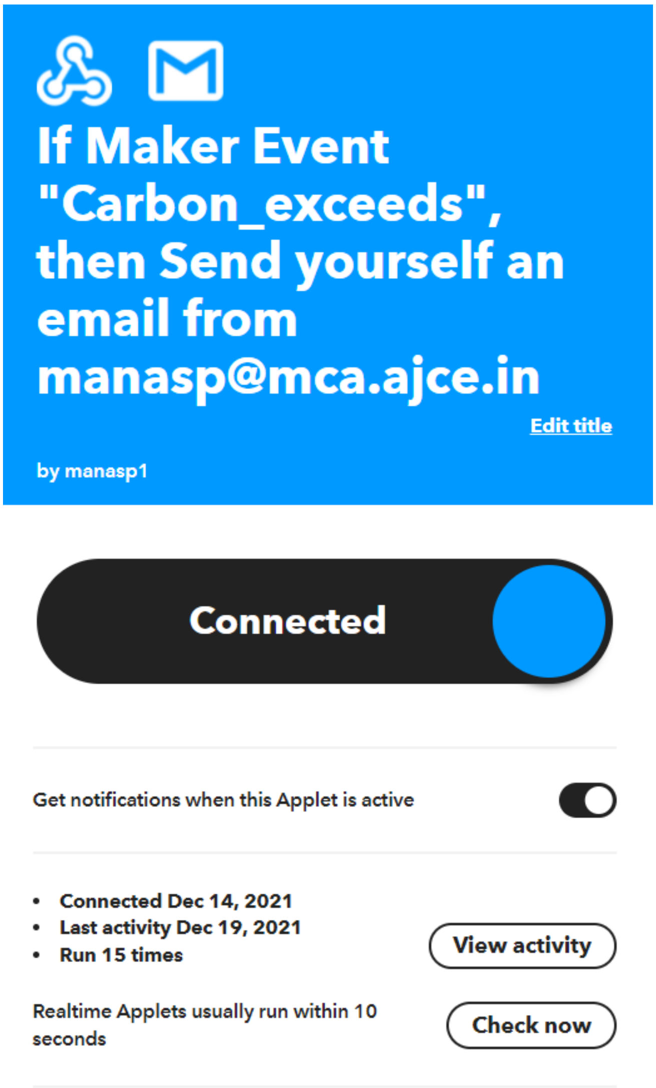

Fig 11. Send mail using webhook

Sending mail to the user when the CO emission exceeds the limit using Webhook provided by IFTTT Applet

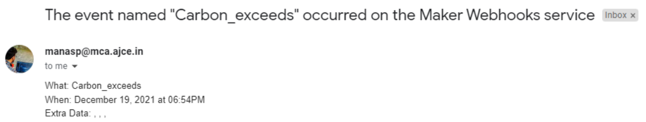

Fig 12. Output when CO Exceeds the limit

V. ANALYZING DATASET

Export dataset in the form of csv from Thingspeak and with the help of Google Colab, an online python editor we can predict and forecast in which year the current vehicle will emit CO beyond the limit using ARIMA (Autoregressive Integrated Moving Averages) and Seasonal ARIMA.

```
[1] import numpy as np
    import pandas as pd
    import statsmodels.api as sm
    import matplotlib.pyplot as plt
    %matplotlib inline

[2] url='https://raw.githubusercontent.com/Manas-P/forecasting_co2_emission_threshold/main/emission_dataset.csv'
    df=pd.read_csv(url)
```

Fig 13. Import modules and dataset

```
[4] df.head()
```

	Date	CO Amount (g/km)
0	2010-01	0.001
1	2010-02	0.026
2	2010-03	0.043
3	2010-04	0.068
4	2010-05	0.089

```
[5] df.tail()
```

	Date	CO Amount (g/km)
127	2020-08	0.641
128	2020-09	0.636
129	2020-10	0.624
130	2020-11	0.612
131	2020-12	0.600

Fig 14. Dataframe head and tail

We have imported the dataset to df variable.
df.head will return first 5 values and df.tail will return last 5 values of the dataset.

DOI:10.5281/zenodo.6365072
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

```
[6] df['Date']=pd.to_datetime(df['Date']) #Converted Month into Datetime
```

```
[7] df.head()
```

	Date	CO Amount (g/km)
0	2010-01-01	0.001
1	2010-02-01	0.026
2	2010-03-01	0.043
3	2010-04-01	0.068
4	2010-05-01	0.089

```
[8] df.set_index('Date',inplace=True) #Set month column as index
```

```
[9] df.head()
```

Date	CO Amount (g/km)
2010-01-01	0.001
2010-02-01	0.026
2010-03-01	0.043
2010-04-01	0.068
2010-05-01	0.089

Fig 15. Customized dataset

Converted date to date time format and assigned Date column as index.

```
[13] ### Testing For Stationarity
     from statsmodels.tsa.stattools import adfuller

[14] test_result=adfuller(df['CO Amount (g/km)'])

[15] #Ho: It is non stationary
     #H1: It is stationary
     def adfuller_test(co):
         result=adfuller(co)
         labels = ['ADF Test Statistic','p-value','#Lags Used','Number of Observations Used']
         for value,label in zip(result,labels):
             print(label+' : '+str(value) )
         if result[1] <= 0.05:
             print("Strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data has no unit root and is stationary")
         else:
             print("Weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary ")

[16] adfuller_test(df['CO Amount (g/km)'])

     ADF Test Statistic : -0.018282603020073354
     p-value : 0.9569937580740829
     #Lags Used : 12
     Number of Observations Used : 119
     Weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary
```

Fig 16. Stationarity test

Stationarity test is done using Dickey Fuller Test and the output came as non-stationarity

```
[17] #Differencing
     df['CO First Difference'] = df['CO Amount (g/km)'] - df['CO Amount (g/km)'].shift(1)

[18] df['CO Amount (g/km)'].shift(1)

     Date
     2010-01-01      NaN
     2010-02-01    0.001
     2010-03-01    0.026
     2010-04-01    0.043
     2010-05-01    0.068
                   ...
     2020-08-01    0.649
     2020-09-01    0.641
     2020-10-01    0.636
     2020-11-01    0.624
     2020-12-01    0.612
     Name: CO Amount (g/km), Length: 132, dtype: float64

[19] df['Seasonal First Difference']=df['CO Amount (g/km)']-df['CO Amount (g/km)'].shift(12)

[21] df.head()
```

Date	CO Amount (g/km)	CO First Difference	Seasonal First Difference
2010-01-01	0.001	NaN	NaN
2010-02-01	0.026	0.025	NaN
2010-03-01	0.043	0.017	NaN
2010-04-01	0.068	0.025	NaN
2010-05-01	0.089	0.021	NaN

```
[22] ## Again test dickey fuller test
     adfuller_test(df['Seasonal First Difference'].dropna())

     ADF Test Statistic : -4.411344479190653
     p-value : 0.00028340703441989907
     #Lags Used : 13
     Number of Observations Used : 106
     Strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data has no unit root and is stationary
```

Fig 17. Differencing

Shift and differencing is done in order to make it stationary.

```
[23] # Auto Regressive modal
     from pandas.plotting import autocorrelation_plot
     autocorrelation_plot(df['CO Amount (g/km)'])
     plt.show()
```

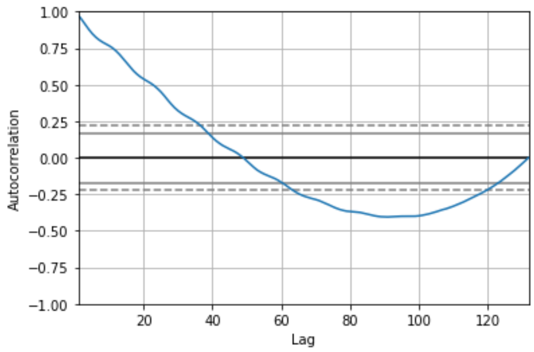

Fig 18. Autocorrelation

```
[24] from statsmodels.graphics.tsaplots import plot_acf,plot_pacf

[25] fig = plt.figure(figsize=(12,8))
     ax1 = fig.add_subplot(211)
     fig = sm.graphics.tsa.plot_acf(df['Seasonal First Difference'].iloc[13:],lags=40,ax=ax1)
     ax2 = fig.add_subplot(212)
     fig = sm.graphics.tsa.plot_pacf(df['Seasonal First Difference'].iloc[13:],lags=40,ax=ax2)

     /usr/local/lib/python3.7/dist-packages/statsmodels/regression/linear_model.py:1358: RuntimeWarning: invalid value encountered in sqrt
       return rho, np.sqrt(sigmasq)
```

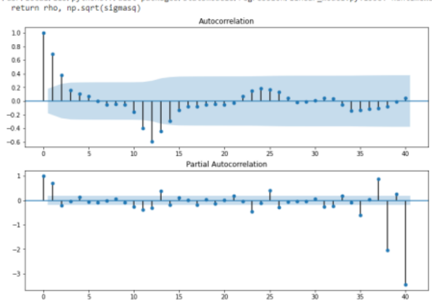

Fig 19. Autocorrelation vs Partial autocorrelation

DOI:10.5281/zenodo.6365072
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

```
[ ] from statsmodels.tsa.arima_model import ARIMA
```

```
[ ] # For non-seasonal data
    model=ARIMA(df['CO Amount (g/km)'],order=(1,1,1))
    model_fit=model.fit()
```

```
[ ] model_fit.summary()
```

ARIMA Model Results

Dep. Variable:	D.CO Amount (g/km)	**No. Observations:**	131
Model:	ARIMA(1, 1, 1)	**Log Likelihood**	370.859
Method:	css-mle	**S.D. of innovations**	0.014
Date:	Sun, 19 Dec 2021	**AIC**	-733.719
Time:	13:13:59	**BIC**	-722.218
Sample:	02-01-2010 - 12-01-2020	**HQIC**	-729.046

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0045	0.000	21.133	0.000	0.004	0.005
ar.L1.D.CO Amount (g/km)	0.8698	0.050	17.360	0.000	0.772	0.968
ma.L1.D.CO Amount (g/km)	-1.0000	0.020	-50.813	0.000	-1.039	-0.961

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.1497	+0.0000j	1.1497	0.0000
MA.1	1.0000	+0.0000j	1.0000	0.0000

```
[ ] df['Forecast']=model_fit.predict(start=112,end=131,dynamic=True)
    df[['CO Amount (g/km)','Forecast']].plot(figsize=(12,8))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1bebdfe990>
```

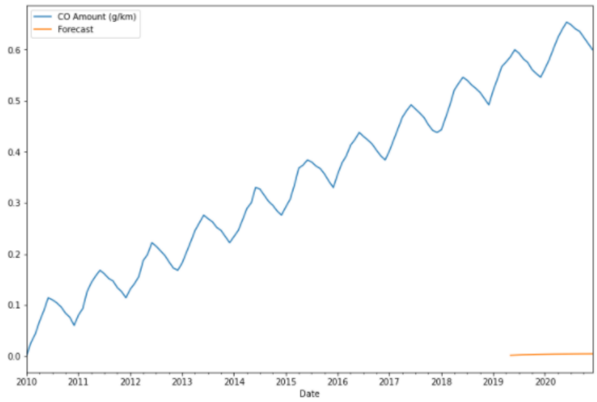

Fig 20. ARIMA for non-seasonal data

ARIMA is implemented for non-seasonal data and since our data is seasonal, the prediction is not valid in the resultant plot

```
[ ] model=sm.tsa.statespace.SARIMAX(df['CO Amount (g/km)'],order=(1, 1, 1),seasonal_order=(1,1,1,12))
    results=model.fit()
```

```
[ ] df['Forecast']=results.predict(start=112,end=131,dynamic=True)
    df[['CO Amount (g/km)','Forecast']].plot(figsize=(12,8))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1be9c1a2d0>
```

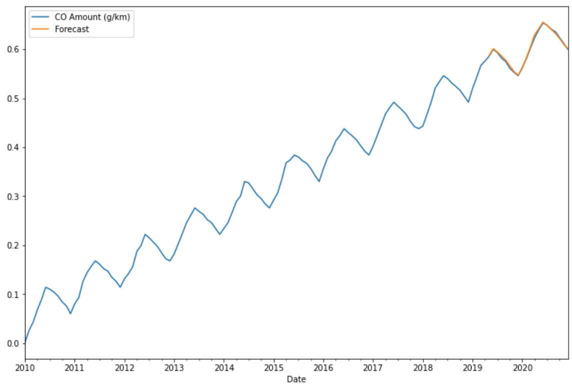

Fig 21. SARIMA for seasonal data

After applying Seasonal-ARIMA, we predicted the correct value and now we can move forward with the forecasting.

```
[ ] #To find future
    from pandas.tseries.offsets import DateOffset
    future_dates=[df.index[-1]+ DateOffset(months=x)for x in range(0,96)] #Given upto 8 years
```

```
[ ] future_datest_df=pd.DataFrame(index=future_dates[1:],columns=df.columns) #Future values
```

```
[ ] future_datest_df.tail()
```

	CO Amount (g/km)	CO First Difference	Seasonal First Difference	Forecast
2028-07-01	NaN	NaN	NaN	NaN
2028-08-01	NaN	NaN	NaN	NaN
2028-09-01	NaN	NaN	NaN	NaN
2028-10-01	NaN	NaN	NaN	NaN
2028-11-01	NaN	NaN	NaN	NaN

Fig 22. Forecast for next 8 years

```
[ ] future_df=pd.concat([df,future_datest_df]) #Concatenate current dataset to future dataset
```

```
[ ] future_df['Forecast'] = results.predict(start = 131, end = 225, dynamic= True)
    future_df[['CO Amount (g/km)', 'Forecast']].plot(figsize=(16, 8))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1be9b08b50>
```

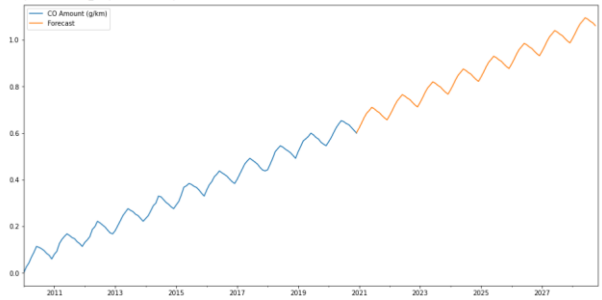

Fig 23. Concatenate current dataset and forecasted dataset

```
[ ] #Predicting the year when the co2 emission will exceed the limit
    df_new = future_df[future_df['Forecast'] >= 1].index[0]
    print("Your vehicle will emit CO beyond the limit at the year: ",df_new)

Your vehicle will emit CO beyond the limit at the year:  2027-04-01 00:00:00
```

Fig 24. Time period when the current vehicle's CO emission exceeds the standards

VI. RESULTS AND DISCUSSIONS

The gas sensor data is analyzed, and an irregular emission is found, which is then reported to the user for further action. The flow diagram of the system's operation is shown in Fig 26. The Arduino uno controller is linked to the Engine Control Unit (ECU), which detects excessive emissions and shuts down the vehicle. This will be performed by removing the spark plug's electrical connection. As a warning to the user, the driver will be allowed a maximum of 200 seconds to park the automobile in a visible safer lane, and the pseudo-code for this specific case is presented below.

```
if(l_gasLevel>1){
   digitalWrite(relayInput, HIGH);
   delay(1000);
   digitalWrite(buzzerPn, HIGH);
 }
 if(l_gasLevel<1){
   digitalWrite(relayInput, LOW);
   delay(1000);
   digitalWrite(buzzerPn, LOW);
 }
```

Fig 25. Pseudo-code for this specific case

DOI:10.5281/zenodo.6365072
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

Meanwhile, the system communicated with the car in regular mode, continually sensing the gas from the exhaust. While the car is functioning normally, the time period remains at zero. The user is notified of the anomalous state in the car.

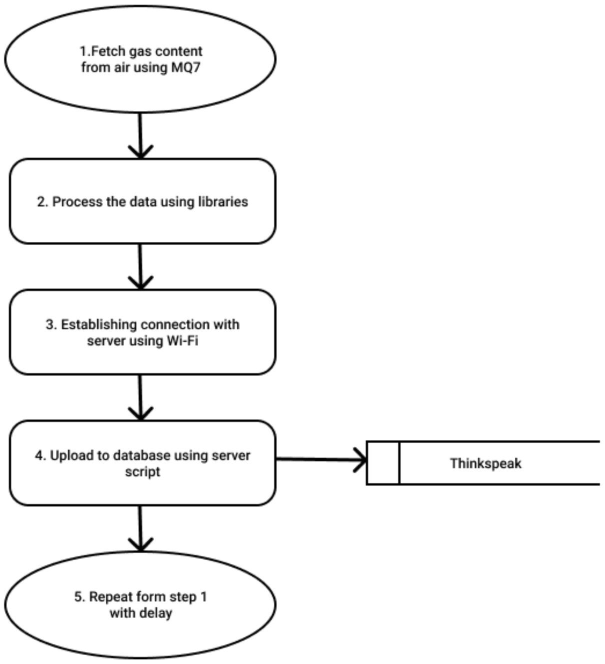

Fig 26. Flow diagram of proposed system

The anomalous emission level is included in the notification. One of the most essential reasons to avoid exhaust emissions is the vehicle's FC expiration. As a result, the proposed system includes special instructions for sending an alarm message and will forecast the period when the vehicle will exceed the limit of CO emission.

VII. Conclusion And Future Scope

Each nation has distinct emission requirements based on its geographical location and available resources, but the majority of countries strive to implement the UNFCCC summit standards in order to minimise emissions and save the environment. The proposed prototype in this research is applied in society and is highly cost effective and takes up considerably less space than existing standards. However, there is one negative circumstance that arises while the car is travelling in elevated mountainous terrain. Because of the need for high pulling torque, standardised emission values in these regions tend to vary. Under these conditions, with the authorization of the government, the procedure of overriding the automated controller takeover may be avoided in order to lessen the likelihood of an accident occurring.

In the future, it will also be necessary to evaluate other gas factors and to upgrade the system to meet the new gas emissions legislation criteria. The device might also recommend nearby approved service outlets for rapid servicing to the user. The suggested automated pollution monitoring system works in tandem with the present trend toward electric cars and hybrid systems to reduce dangerous gas emissions into the atmosphere. In order to properly maintain the vehicle, the government standards for identifying expired and summoned cars must be rigorously observed.

VIII. Reference

1. Ma, Y., Richards, M., Ghanem, M., Guo, Y. and Hassard, J., 2008. Air pollution monitoring and mining based on sensor grid in London. Sensors, 8(6), pp.3601-3623.
2. Khedo, K.K., Perseedoss, R. and Mungur, A., 2010. A wireless sensor network air pollution monitoring system. arXiv preprint arXiv:1005.1737.
3. Cordova-Lopez, L.E., Mason, A., Cullen, J.D., Shaw, A. and Al Shamma'a, A.I., 2007. Online vehicle and atmospheric pollution monitoring using GIS and wireless sensor networks. In Journal of Physics: Conference Series (Vol. 76, No. 1, p. 012019). IOP Publishing.
4. Su, J.G., Jerrett, M., Beckerman, B., Wilhelm, M., Ghosh, J.K. and Ritz, B., 2009. Predicting traffic-related air pollution in Los Angeles using a distance decay regression selection strategy. Environmental research, 109(6), pp.657-670.
5. Riahi, K., Rao, S., Krey, V., Cho, C., Chirkov, V., Fischer, G., Kindermann, G., Nakicenovic, N. and Rafaj, P., 2011. RCP 8.5 A scenario of comparatively high greenhouse gas emissions. Climatic Change, 109(1-2), p.33.

DOI:10.5281/zenodo.6365072
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam