IoT based Air Quality monitoring system using MQ135&MQ7 with Machine Learning analysis

Kinnera Bharath Kumar Sai
M.Tech CSE
VIT University, Vellore

Subhaditya Mukherjee
B.Tech CSE
VIT University, Vellore

Dr. Parveen Sultana H
Associate Professor
Department of CSE, VIT University

***Abstract*—This paper deals with measuring Air Quality using MQ135 sensor along with Carbon Monoxide CO using MQ7 sensor. Measuring Air Quality is an important element for bringing lot of awareness in the people to take care of the future generations a healthier life. Based on this, Government of India has already taken certain measures to ban 'Single Stroke' and 'Two Stroke' Engine based motorcycles which are emitting high pollution comparatively. We are trying to implement the same system using IoT platforms like Thingspeak or Cayenne, we can bring awareness to every individual about the harm we are doing to our environment. Already, New Delhi is remarked as the most pollution city in the world recording Air Quality above 300PPM. We have used easiest platform like Thingspeak and set the dashboard to public such that everyone can come to know the Air Quality at the location where the system is installed. Also, we have reduced the cost of components used on comparing with the papers referred.**

***Index Terms*—IoT, MQ135, MQ7, Thingspeak, Machine Learning**

I. Introduction

Air is getting polluted because of release of toxic gases by industries, vehicle emissions and increased concentration of harmful gases and particulate matter in the atmosphere. The level of pollution is increasing rapidly due to factors like industries, urbanization, increasing in population, vehicle use which can affect human health. Particulate matter is one of the most important parameter having the significant contribution to the increase in air pollution [2]. This creates a need for measurement and analysis of real-time air quality monitoring so that appropriate decisions can be taken in a timely period. This paper presents a real-time standalone air quality monitoring. Internet of Things (IoT)is nowadays finding profound use in each and every sector, plays a key role in our air quality monitoring system too. The setup will show the air quality in PPM in webpage so that we can monitor it very easily. In this IoT project, you can monitor the pollution level from anywhere using your computer or mobile [1].

II. Literature Survey

As shown in Table 1 cited at [11], it explains about the Air Quality Index ranges. Firstly, 0-50 PPM can be considered completely safe. 51-100 PPM can be considered as Moderate where this could be usually observed at traffic areas [13]. 100-150 PPM can be considered as Unhealthy but only for sensitive groups. Above 151 PPM [11]is completely unsafe or unhealthy where India capital New Delhi falls in this range. Its very rare to record 300 PPM and above which can be considered as Hazardous, possibly due to Coal gas in mines [10].

TABLE I
Air Quality index

Range(PPM)	Status
0-50	Good
51-100	Moderate
100-150	Unhealthy for sensitive groups
151-200	Unhealthy
201-300	Very Unhealthy
301-500	Hazardous

This paper referred the idea cited at [1] with less cost i.e., while pushing the data to the cloud, no need to see the output on LCD which adds more cost to the project [14] [18].On targeting IoT as a platform, our intension should be to present the idea on internet using the platforms like thinger.io or thingspeak or Cayenne website which are beautifully designed to present the output and even able to download the dataset. When doing an experiment air quality monitoring, no need to use LPG or methane detecting sensors as it is used for Home/office safety. This paper used WiFi to push the data onto the cloud rather using GSM or GPRS module [2]. The problem in another paper that cited at [3] hasn't calibrated the sensor and not converted the sensor output value into PPM. As per the guidelines by UN Data, 0-50 PPM is SAFE value, 51-100 is moderate as shown in Table 1. New Delhi, the capital of India is the most polluted city in the world recorded around 250PPM [15]. As this paper uses two sensors, both of them have internal heater element, it draws more power(P=V*I), so though the both sensors are turned ON, its output voltage levels varies and shows unpredictable values due to insufficient power drive. So we used a 9Volts battery and a 7805 family LM7805 Regulator for the CO sensor MQ7 as power from Arduino alone is not sufficient to drive two sensors.

The paper cited at [4] is not clear with the components used and the cloud used.This paper also aims to implement the machine learning on the real time dataset collected from the Thingspeak website and try to convey the information about the adverse affects on one's health to the people and government if the same pollution continues. This paper also aims to extend by adding three more sensors related to air quality… i.e, ozone sensor, PM 2.5 laser dust sensor, MG811 (CO_2) sensor that acts as an all-in-one setup giving in depth monitoring of the air quality [4]. The paper cited at [5] had completely taken wrong assumption where they have showed the output 997PPM as the fresh air, where Delhi which is the most polluted city recording 250PPM. Its clear understanding that they haven't calibrated the sensor and didn't even convert the raw sensor data into [16]PPM using derivations we did. They have used LocalHost which is limited where they are able to see the output only on the laptop within the area where experimental setup is connected. But this paper targets using standard IoT platform which is highly secured and open source [6].

III. Procedure

We have used Arduino Uno Development kit that comes with ATMega328P microcontroller. In order to provide WiFi Support for it, we have used cost effective ESP-01 WiFi module which helps us to connect to the ThingSpeak Platform. Figure 1 represents the connections between the components used like Arduino Uno, MQ135, MQ7, ESP-01 Wifi Module, 9Volts Battery, LM7805 Regulator. From Figure 1, ESP-01

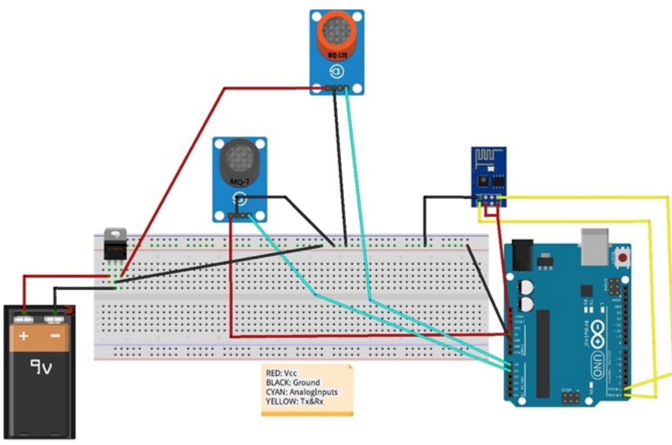

Fig. 1. Connections diagram

is connected to 3.3Volts pin of Arduino Uno. MQ135 is connected to 5Volts pin of Arduino Uno. As power wont be sufficient to drive one more sensor, MQ7 is connected to 9Volts Battery via 5Volts LM7805 Regulator. ESP-01 is connected to the Local Hotspot by giving corresponding SSID and Password. The reason for using LM7805 Regulator is that 9Volts supply should not be directly [17]given to MQ7 sensor where it needs only 5Volts input at maximum, so regulator does the job of stepping 9Volts to 5Volts [7] [9].

The most important step is to calibrate the sensor in fresh air and then draws an equation that converts the sensor output voltage value into our convenient units PPM (parts per million). Here are the mathematical calculations derived which is cited at [12].

Fig 5: Internal circuit diagram of MQ135 sensor R_s and R_l combined From Ohm's Law, at constant temperature, we can derive I as follows:

$$I = \frac{V}{R} \tag{1}$$

From fig 5, equation 1 is equivalent to [18]

$$I = \frac{V_c}{R_s + R_l} \tag{2}$$

From Figure 2, we can obtain the output voltage at the load resistor using the value obtained for I and Ohm's Law at constant temperature. V = I*R

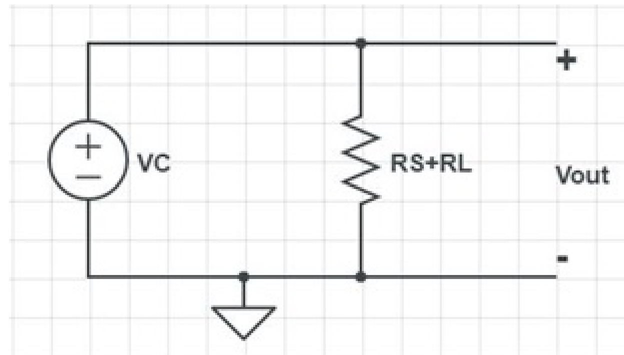

Fig. 2. Internal circuit diagram of MQ135

$$V_{R_l} = [\frac{V_c}{R_s + R_l}] * R_L \tag{3}$$

$$V_{R_l} = [\frac{V_c * R_l}{R_s + R_l}] \tag{4}$$

So now we solve for R_s:

$$V_{R_l} * (R_s + R_l) = V_c * R_L \tag{5}$$

$$(V_{R_l} * R_s) + (V_{R_l} * R_l) = V_c * R_L \tag{6}$$

$$V_{R_l} * R_s = (V_c * R_l) - (V_{R_l} * R_l) \tag{7}$$

$$R_s = \frac{(V_c * R_l) - (V_{R_l} * R_l)}{V_{R_l}} \tag{8}$$

$$R_s = \frac{(V_c * R_l)}{V_{R_l}} - R_l \tag{9}$$

Equation 9 help us to find the internal sensor resistance for fresh air [12].

From the graph shown in fig 3, we can see that the resistance ratio in fresh air is a constant:

$$\frac{R_s}{R_0} = 3.6 \tag{10}$$

Value 3.6 which is mentioned at Equation 10 is depicted from the datasheet shown in Fig 3. To calculate R_0, we will need to find the value of the R_s in fresh air. This will be done by taking the analog average readings from the sensor and converting it to voltage [12]. Then we will use the R_s formula to find R_0. First of all, we will treat the lines as if they were linear. This way we can use one formula that linearly relates the ratio and the concentration [19] [20]. By doing so, we can find the concentration of a gas at any ratio value even outside of the graph's boundaries. The formula we will be using is the equation for a line, but for a log-log scale. The formula

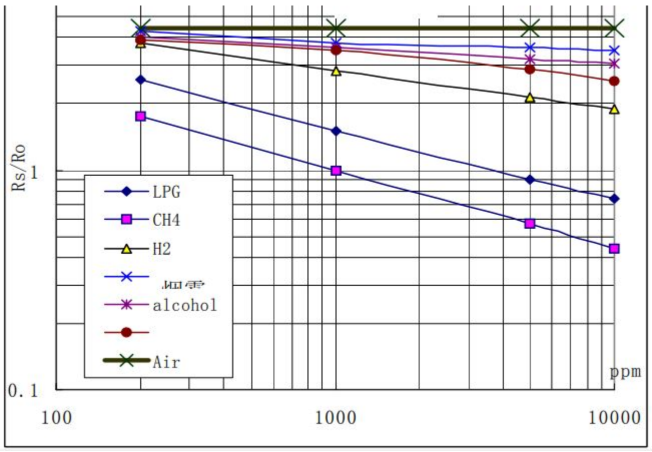

Fig. 3. : MQ135 Datasheet-Change in Resistance vs change in PPM

for a line is [12]: From above Figure 3, we try to derive the following calculations.

$$y = mx + b \tag{11}$$

For a log-log scale, the formula looks like this:

$$log_{10}y = m * log_{10}x + b \tag{12}$$

Let's find the slope. To do so, we need to choose 2 points from the graph. In our case, we chose the points (200,2.6) and (10000,0.75) from the LPG (Liquified Petroleum gas)line from fig 3. The LPG line is a result of sensor under testing with various level of LPG as input. The formula to calculate slope m(here) is the following:

$$m = \frac{logy - log(y_0)}{logx - log(x_0)} \tag{13}$$

If we apply the logarithmic quotient rule we get the following:

$$m = \frac{log(y/y_0)}{log(x/(x_0)} \tag{14}$$

Now we substitute the values for x, x_0, y, and y_0:

$$m = \frac{log(0.75/2.6)}{log(10000/200)} \tag{15}$$

$$m = -0.318 \tag{16}$$

Now that we have m, we can calculate the y intercept. To do so, we need to choose one point from the graph (once again from the CO2 line). In our case, we chose (5000,0.9)

$$log(y) = m * log(x) + b \tag{17}$$

$$b = log(0.9) - (-0.318) * log(5000) \tag{18}$$

$$b = 1.13 \tag{19}$$

Now that we have m and b, we can find the gas concentration for any ratio with the following formula:

$$log(x) = \frac{log(y) - b}{m} \tag{20}$$

However, in order to get the real value of the gas concentration according to the log-log plot we need to find the inverse log of x:

$$x = 10^{\frac{log(y)-b}{m}} \tag{21}$$

Using equations 9 and 21, we will be able to convert the sensor output values into PPM (Parts per Million) [16]. Now we developed the Code and flashed into the Arduino Uno giving proper connections as mentioned in fig 1.

IV. Results

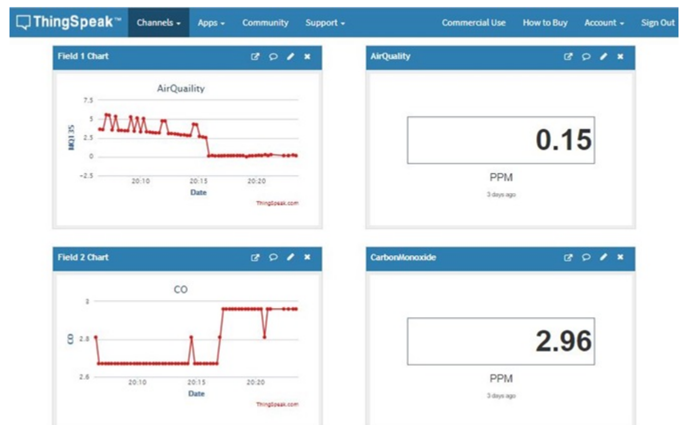

Fig. 4. Output on Thingspeak

Fig. 5. Graph showing AirQuality

After connecting the ESP-01 successfully to the hotspot, it gets established with Thingspeak wesbite and the account API Key is written in Arduino Code which helps to save the data only to our account bearing the given API key. Thingspeak needs 15 seconds of refresh interval to push to the data. Fig 4 shows the field charts of MQ135 and MQ7 sensor values for the location where the experiment is conducted in PPM (Parts per million) [7] [8]. Also figure 4 shows the visualization charts for corresponding sensors.

Fig 5 showing the graphical analysis of the values collected with timestamping on X axis and Air Quality PPM on Y axis.

For analysis, we use Jupyter Notebook hosted on Google Co-Laboratory. A GPU was used to train the model and reduce computation time. The data was collected by combination of four sensors kept in and around the University for a few days. Continuous data was collected before saving it to a storage device over the cloud. For any dataset, pre-processing is the most important step [21]. The first step was to parse the timestamp generated from the device and make it into a format usable by the model. This was done using the following code. The second step was to remove outliers from the data and fill missing values. This was done by replacing the missing value with a value from the same column which would ensure that the values did not get skewed completely. The third was to remove outliers. From data visualization, it was seen that there was a recurring value -999 which could be the sensor output when the sensor was switched off and these redundant values were removed [22].

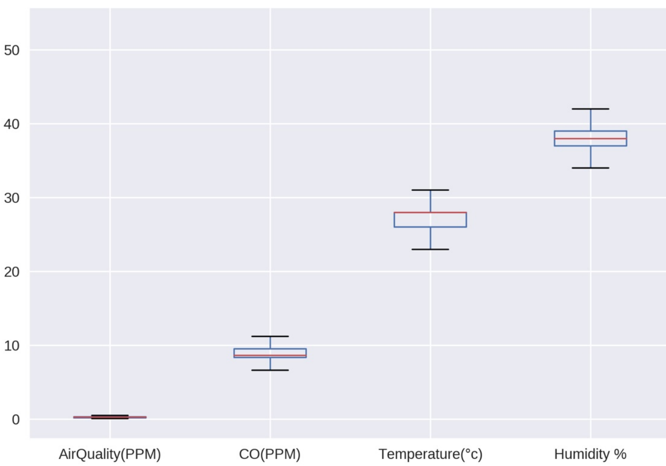

Fig. 6. Box Plot

From the plot shown in fig 6, we can easily identify the median of values present in the dataset. This is then used for further observation. We can see that the values of air quality is between 0 to 10 PPM which is safe from Table 1. The values of CO are around 9PPM which is a little high but people will not suffer from any adverse effects. The temperature also has value between 20-30 and humidity percentage is around 40%

From the plot shown in fig 7 we can easily identify the correlation of the parameters in the data set. It is easy to see that air quality is dependent on these factors to various extents. A high negative as well as a moderate correlation is present. This is an important source of information. From it we can identify that Air Quality is inversely related to the rest of the values of Humidity, Temperature. CO values are highly negatively related to Temperature as well. It is to be noted that negative relations mean that as one value decreases, the other increases.

From Histogram plot shown in fig 8, we can obtain maximum and minimum values over a period for every data point. Most of the Air Quality values are between 0 and 1, which indicate not very high values of pollution. For the values of Carbon Monoxide, most of the values are between 7.5 which also indicates moderate values of CO. The entry id Histogram

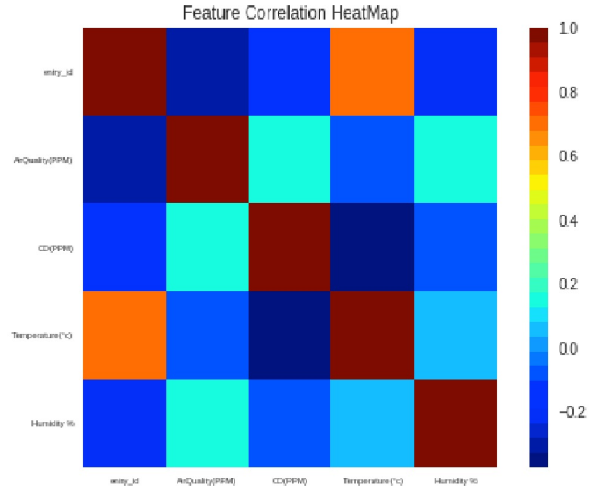

Fig. 7. Correlation Heat map

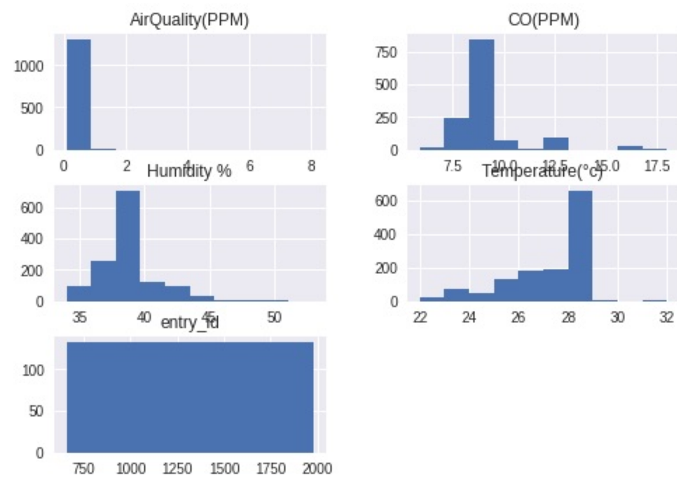

Fig. 8. Histogram of the dataset

can be ignored as it is of no significance. The temperature values are also mostly between 28 and 30 degrees which indicated room temperature. As for humidity %, most of the values seem to be between 35 and 40 which seems to be little high.

From the plot shown in fig 9 we can easily see the data for every data point over a period of time. This is done after outliers are removed so it becomes easy for us to draw conclusions from this data. Over the period of time we can see that the Air Quality values were the highest on some days but dipped down and varied a lot initially. This coincided with the dipping of Temperature values which is a major point to note. The CO values did not change much but considerably varied with variation of Temperature which is also a point to be noted. As for the Humidity values, they do not seem to have a lot of relation with the rest of the variables.

From the plot shown in fig 10, we can easily see the data as an area under the curve for every data point over a period of

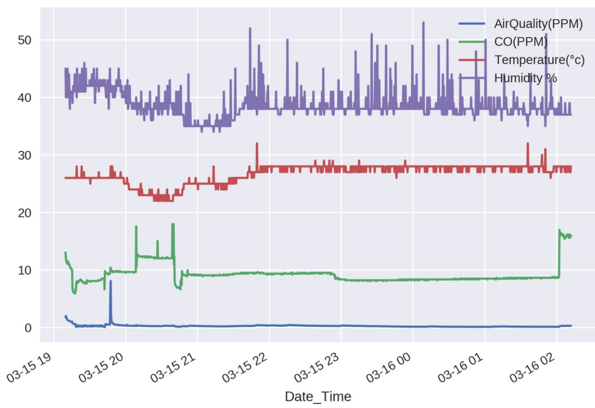

Fig. 9. Line Plot of the data

time. This enables us to visualize how the data is represented over time and perform data analysis. The area plot mostly gives the same data as the line plot and no additional information is clear.

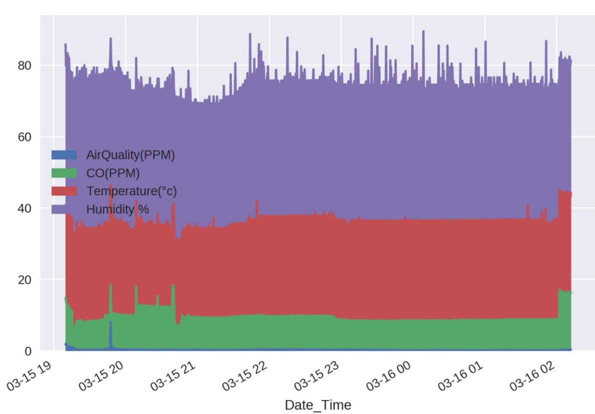

Fig. 10. Area Plot of the data

The root mean square error for every data point is as follows

- rmse value for AirQuality(PPM) is 0.08274985657405932
- rmse value for CO(PPM) is 2.8055052798658817
- rmse value for Temperature(c) is 1.4761492661822175
- rmse value for Humidity is 1.706479462754771

This indicates a relatively good performance of the model. Especially for the Air Quality Values.

V. Conclusion

From all the above information provided, we are able to calculate AirQuality in PPM. The problem with MQ135 sensor is that specifically it can't tell the Carbon Monoxide or Carbon Dioxide level in the atmosphere, but the pros of MQ135 is that it is able to detect smoke, CO, CO_2, NH_4 as mentioned in figure 3. So, just to tell the individual gases level particularly, we have used CO (Carbon Monoxide) MQ7 sensor. This paper also corrects the PPM calculations mentioned at Literature Survey. This project can be used both for indoor as well as outdoor. For indoor, we can make this kit as a compact device such that if every home started using the device, we can monitor the indoor air quality of a particular targeted area. Due to increasing air pollution, there is necessity to keep an eye on Indoor air quality too. But for outdoor purpose, certainly one sensor is not sufficient because one sensor has a sensitivity range of around 1 meter, so a network of sensors has to be deployed to monitor the outdoor air quality. Enough care is taken while calibrating the sensors. This paper also targets the Machine Learning analysis on the dataset collected.

Training the model considers the 'Vector Auto Regression model' was found to be the best choice to train this model. A VAR model is used for multivariate time series analysis. The whole structure is that the variable is a linear function of the past lags of the current variable with the past lags of the other variables. For VAR(1), each variable is a linear function of lag1 values and so on [22]. Such a model in general implies that every variable depends on every other variable and thus the VAR model in the end can be written as a series of individual models. The VAR model can also be estimated by estimating each equation separately. Many models were tested and even an Long Short Term Memory Model(LSTM) was considered and it was found that the VAR gave the best results in the form of root mean square error.

VI. Future Work

We can use one more sensor that tells the ozone layer status, but it costs very high. Also we can use PM2.5 laser dust sensor helpful for exclusively for vehicle and factory emissions sensing. Thingspeak has a limitation that it requires 15-20 seconds for every push of the values which is not reliable. We plan to use another IoT platform mydevices cayenne which is very fast in showing the values from the Arduino that helps us to collect more values in the dataset. Cayenne also comes with a ready android/ios application. But it doesn't work with Arduino Uno rather works with only NodeMCU or raspberry Pi. If we use NodeMCU, even the cost becomes less than the current setup. But the limitation in NodeMCU is, it has only one analog input pin, so we will use ADS1115 I2C 16Bit ADC as an analog extender for NodeMCU or a simple CD4051 8 to 1 Analog Multiplexer could easily overcome the problem of having only one analog input pin of NodeMCU(ESP8266). CD4051 Multiplexer is highly recommended as it is very cheap to purchase and easy to handle many sensors for NodeMCU. NodeMCU (ES8266) has inbuilt Wi-fi support (ESP-12E) and microcontroller. We can link this to the Facebook API using IFTTT, Webhooks and adafruit platform collectively, such that user can request the airquality via facebook messenger chat application and get the output on the screen using chatbot. Machine learning can also be implemented on the dataset such that we can predict the harmfulness of the airquality on people if the same bad airquality continues.

REFERENCES

[1] Tragos, E. Z., Angelakis, V., Fragkiadakis, A., Gundlegard, D., Nechifor, C. S., Oikonomou, G., ... & Gavras, A. (2014, March). Enabling reliable and secure IoT-based smart city applications. In 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS) (pp. 111-116). IEEE.

[2] Shah, J., & Mishra, B. (2016, January). IoT enabled environmental monitoring system for smart cities. In 2016 International Conference on Internet of Things and Applications (IOTA) (pp. 383-388). IEEE.

[3] Pasha, S. (2016). ThingSpeak based sensing and monitoring system for IoT with Matlab Analysis. International Journal of New Technology and Research, 2(6).

[4] Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012, December). Future internet: the internet of things architecture, possible applications and key challenges. In 2012 10th international conference on frontiers of information technology (pp. 257-260). IEEE.

[5] "Malaya Ranjan, Rai kumar, "Understanding Parts per million in real time air quality index", Journal of Mathematics and advanced sciences, pp. 23-29, September 2009.

[6] Kumar, N. S., Vuayalakshmi, B., Prarthana, R. J., & Shankar, A. (2016, November). IOT based smart garbage alert system using Arduino UNO. In 2016 IEEE Region 10 Conference (TENCON) (pp. 1028-1034). IEEE.

[7] Kumar, S., & Jasuja, A. (2017, May). Air quality monitoring system based on IoT using Raspberry Pi. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 1341-1346). IEEE.

[8] Talari, S., Shafie-Khah, M., Siano, P., Loia, V., Tommasetti, A., & Catalão, J. (2017). A review of smart cities based on the internet of things concept. Energies, 10(4), 421.

[9] Ma, Y., Yang, S., Huang, Z., Hou, Y., Cui, L., & Yang, D. (2014, December). Hierarchical air quality monitoring system design. In 2014 International Symposium on Integrated Circuits (ISIC) (pp. 284-287). IEEE.

[10] Ahlgren, B., Hidell, M., & Ngai, E. C. H. (2016). Internet of things for smart cities: Interoperability and open data. IEEE Internet Computing, 20(6), 52-56.

[11] Cho, R. (2018, June 26). What you should know about air quality alerts. Retrieved from https://phys.org/news/2018-06-air-quality.html

[12] Systems, J. (2016, May 09). Understanding a Gas Sensor. Retrieved from https://jayconsystems.com/blog/understanding-a-gas-sensor

[13] Xiaojun, C., Xianpeng, L., & Peng, X. (2015, January). IOT-based air pollution monitoring and forecasting system. In 2015 International Conference on Computer and Computational Sciences (ICCCS) (pp. 257-260). IEEE.

[14] Fang, S., Da Xu, L., Zhu, Y., Ahati, J., Pei, H., Yan, J., & Liu, Z. (2014). An integrated system for regional environmental monitoring and management based on internet of things. IEEE Transactions on Industrial Informatics, 10(2), 1596-1605.

[15] Zheng, K., Zhao, S., Yang, Z., Xiong, X., & Xiang, W. (2016). Design and implementation of LPWA-based air quality monitoring system. IEEE Access, 4, 3238-3245.

[16] Rushikesh, R., & Sivappagari, C. M. R. (2015, October). Development of IoT based vehicular pollution monitoring system. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 779-783). IEEE.

[17] Marques, G., & Pitarma, R. (2016). An indoor monitoring system for ambient assisted living based on internet of things architecture. International journal of environmental research and public health, 13(11), 1152.

[18] Manna, S., Bhunia, S. S., & Mukherjee, N. (2014, May). Vehicular pollution monitoring using IoT. In International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014) (pp. 1-5). IEEE.

[19] Kang, B., Park, S., Lee, T., & Park, S. (2015, January). IoT-based monitoring system using tri-level context making model for smart home services. In 2015 IEEE International Conference on Consumer Electronics (ICCE) (pp. 198-199). IEEE.

[20] Ibrahim, M., Elgamri, A., Babiker, S., & Mohamed, A. (2015, October). Internet of things based smart environmental monitoring using the Raspberry-Pi computer. In 2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC) (pp. 159-164). IEEE.

[21] Kök, İ., Şimşek, M. U., & Özdemir, S. (2017, December). A deep learning model for air quality prediction in smart cities. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 1983-1990). IEEE.

[22] Jo, B., & Khan, R. (2018). An Internet of Things System for Underground Mine Air Quality Pollutant Prediction Based on Azure Machine Learning. Sensors, 18(4), 930.