How to Cite:
Nagabharathi, & Dhinnesh, N. A. D. C. (2022). IoT enabled air quality meter with digital dashboard. *International Journal of Health Sciences*, *6*(S3), 1956–1963.
https://doi.org/10.53730/ijhs.v6nS3.5904

IoT enabled air quality meter with digital dashboard

Nagabharathi
PG Student, MCA, MepcoSchlenk Engineering College, Sivakasi, India
Email: nagabharathiparamasivam@gmail.com

Navin Dhinnesh A. D. C
Assistant Professor, MCA, MepcoSchlenk Engineering College, Sivakasi, India
Email: navindhinneshadc@mepcoeng.ac.in

***Abstract*---**Now a day there are lot of pollution around us. They cause lot of diseases in our day-to-day life. Some of them are asthma, lung cancer and some chronic diseases. Adding to that Ozone layer is getting damaged due to pollution. It could be useful if we could monitor the pollution occurring near our home, so that the pollution can be controlled if the pollution level is high. This project is very much important because it is very much related to human lives. Since many diseases occur due to pollution, there is a needy that we must control the pollution occurring near our home.

***Keywords*---**Arduino UNO R3, ultrasonic sensor, IR Sensor, GPS & GSM, buzzer.

Introduction

The biggest problem of every developed and developing nation is air pollution. Due to industrialization and raise of usage of vehicles viz two-wheelers three-wheelers and four wheelers especially in urban areas leads to severe health issues. By monitoring the level of air pollution through IoT based air pollution monitoring meter using a webserver we can control the pollution level over an area. In this project we can monitor the number of harmful gases present in the air like CO2, smoke, benzene, alcohol, NH3, LPG and NO2 and can measure their amount accurately from anywhere using a PC or mobile phone. Today I have an opportunity to build an environment without air pollution which could save a number of lives by identifying the amount of pollution prevailing are in an area using IoT enabled air quality meter. Though a large data is accumulated only a small area is useful and thus managing the accumulated data is a big challenge. IoT systems are prone to cyber-attacks and malware.

International Journal of Health Sciences ISSN 2550-6978 E-ISSN 2550-696X © 2022.
***Corresponding author**: Nagabharathi; Email: nagabharathiparamasivam@gmail.com*
Manuscript submitted: 18 Nov 2021, Manuscript revised: 09 Feb 2022, Accepted for publication: 27 March 2022

Materials and Methods

User requirements

As the name suggests User requirements are the requirements of the users. These are the requirement demanded by the end-user. This requirement shows how the prototype/product should perform in terms of the product produced and the conditions the product should be made the main scope of this project is to detect the amount of pollution causing pollutants in the environments and analyze it using an application and try to reduce the amount of such substances.

Non-functional requirements

Non-functional requirements are otherwise called NFR that define the attributes like performance, maintainability, usability, scalability. They serve two important aspects such as usability and effectiveness of the entire system. Some of the NFR includes Product family-oriented attributes, Reusability, Modifiability that is addition of new features into the already existing system. The system must be smaller and compact in size and should consume lesser power. It should contain products which are easily available in market so that if it gets failed it can be replaced. The prototype is exposed to harmful gases which should resist it to certain level. Subjective characterization must be avoided.

Hardware description and schematics

The hardware mainly consists of 1. MQ135 sensor, jumper wires, 2. breadboard, 3. ESP8266 for data collection. A Computer with minimum processor 4. (INTEL core 2.00GHZ), 5. 64-bit OS, x-64 based processor RAM of 4 GB.

MQ135-Gas sensor

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.

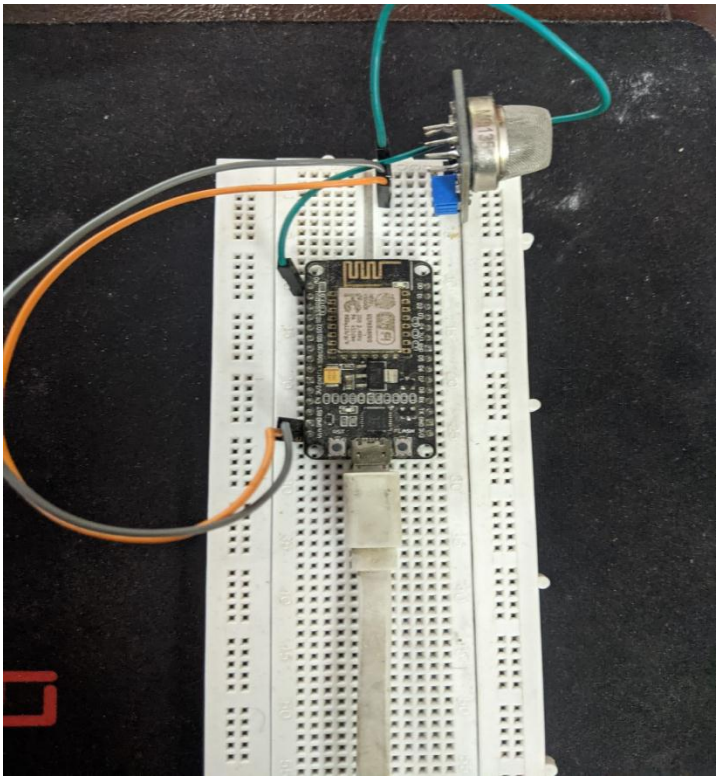


Figure 1. Components installation insider enclosure of prototype.

ESP8266-Wi-Fi module

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability as a Wi-Fi Shield offers. The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community.

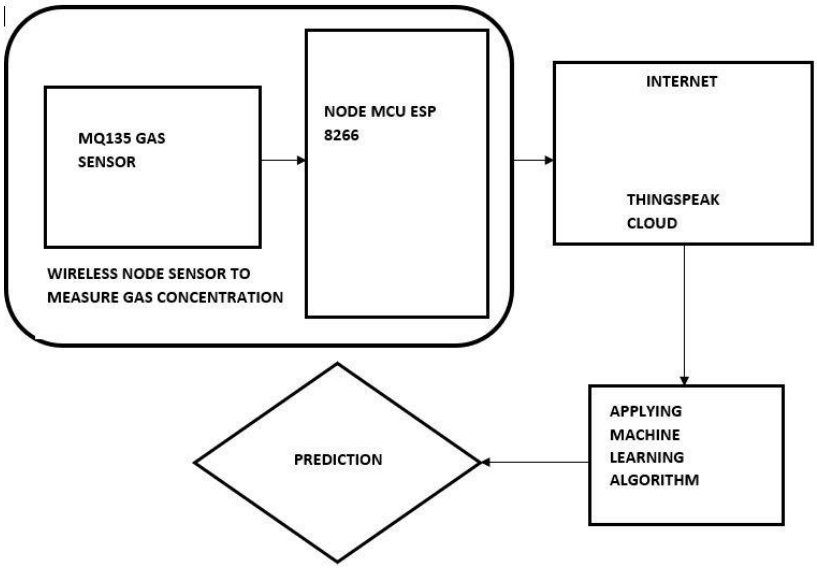


Figure 2. System architecture

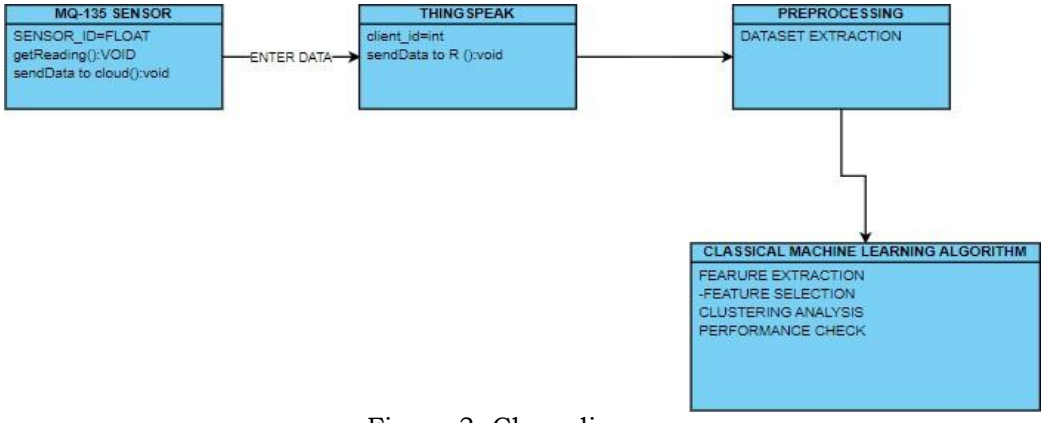


Figure 3. Class diagram

Module design

The MQ-135 sensor is connected with ESP8266 using jumper wires and the code is injected into the system. The sensor starts sensing data and sends to the cloud via ESP8266 WI-FI module. There in the cloud the data is collected and visualized according to pre-defined settings set by the user and can be seen anywhere around the world. From there the data collected is analyzed using an application called as R studio and visualized according to the Machine learning algorithm. Connect MQ-135 sensor with ESP8266 with the help of jumper wiresand breadboard. Connect the pins A0 pin to A0 of the Node MCU and ground pins to the ground. Then with the help of USB cable connect it with the PC/Laptop to importcode. Write the code that is required to run the sensor to collect the data and send it to the cloud and save it. After saving it compile and run the import the code into the Node MCU. Run the serial monitor in the Arduino IDE and we

can see the data being sent to the Thing Speak server cloud. It's possible only when we initialize the connection of the Node MCU. Setting up a WI-FI hotspot and password is necessary. Save the code so that Node MCU can automatically connect next time automatically. Open Thing Speak and setup an account. Using the Write Key as a verification for confirming the user login. Once the setup process is complete data can be viewed and downloaded in the form of .csv file and can be analyzed. The collected data is raw and some data may be missing. So, it must be pre-processed before using it for the purpose of analysis. The preprocessed data from the previous module is used for performing algorithms like K-means clustering and Hierarchical clustering and the result is analyzed. The result that is got in the previous model is visualized in the form of graphs and other graphical forms for our easy understanding.

Analyzing dataset

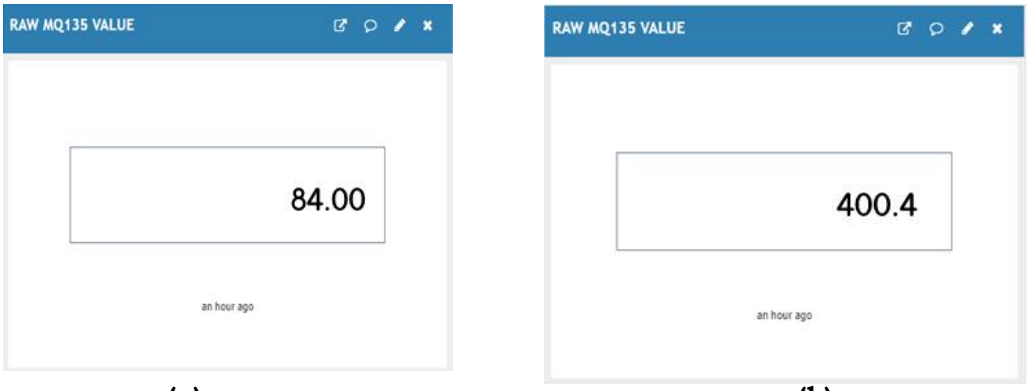
Various clustering techniques are used for analyzing the dataset. They are as follows:

- **K-Means clustering**
  Clustering is an unsupervised learning technique. It's the task of grouping along a group of objects in an exceedingly method that objects within the same cluster area unit a lot of just like one another than to things in different clusters. Similarity is associate degree quantity that reflects the strength of relationship between 2 data objects. K-means bunch is that the most typically used unsupervised machine learning algorithmic program for dividing a given dataset into k clusters. Here, k represents the number of clusters and should be provided by the user.
- **K-Means clustering using air quality dataset**
  Here, we used Air Quality sensor to collect data. Therefore, once we prepared the dataset the following process takes place:
  - Load the view the dataset.
  - Pre-process the dataset.
  - Apply K-Means clustering algorithm.
  - Visualize clustering results.
- **Hierarchical clustering using air quality dataset**
  Hierarchical clustering: Hierarchical clustering is referred to as hierarchal cluster analysis, is an associate degree rule that teams similar objects into teams referred to as clusters. The termination may be a set of clusters, wherever every cluster is distinct from the opposite cluster, and also the objects inside every cluster square measure loosely like one another. Hierarchical Clustering using Air Quality Dataset.
  - Load and view the dataset
  - Pre-process the dataset
  - Scale it
  - Find the distance

  The result is visualized in the form of dendrogram.

Results and Discussions

Finding whether its distinct or not we often use empirical judgement. Cluster plot is used and we have a larger dataset the plot is so clutter and therefore we evaluate cluster centers and found that they are further apart and we can clearly say that they are distinct. Each leaf corresponds to at least one observation. As we tend to move up the tree, observations that area unit kind of like one another area unit combined into branches, that area unit themselves coalesced at the next height. it is easier to choose on the quantity of clusters by looking at the dendrogram. There are 987 number of correctly classified instances and there 21 number of instances which are not correctly classified.



(a) **(b)**

Figure 4. (a) Shows normal gas concentration in Thing-Speak server and (b) shows abnormal gas concentration in Thing-Speak server

Accuracy

Total no of correctly classified instances = 987
Total no of incorrectly classified instances = 21
Accuracy = 987/ (987 + 21)
= 0.979
Therefore, our model has achieved 97 % accuracy

Precision

Precision is defined as the no of true positives divided by sum of true positives and false positives.
Precision = TP/ (TP + FP)

Visualizing

A. Fivz cluster

Provides ggplot2-based elegant visualization of partitioning methods including kmeans [stats package]; pam, clara and fanny [cluster package]; dbscan [fpc package]; Mclust [mclust package]; HCPC [FactoMineR]; hkmeans [factoextra]. Observations are represented by points in the plot, using principal components if ncol(data) An ellipse is drawn around each cluster.

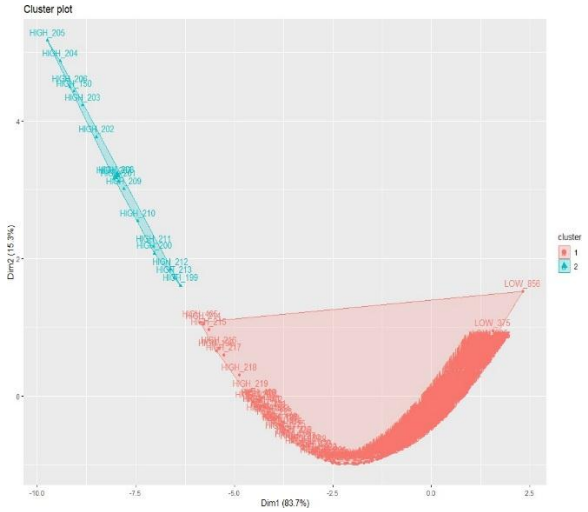


Figure 5. Visualize FIVZ cluster plot of partitioned data

B. Dendrogram

Dendrogram(tree) generates a dendrogram plot of the hierarchical binary cluster tree. A dendrogram consists of many *U*-shaped lines that connect data points in a hierarchical tree. The height of each *U* represents the distance between the two data points being connected.

If there are 30 or fewer data points in the original data set, then each leaf in the dendrogram corresponds to one data point.

If there are more than 30 data points, then dendrogram collapses lower branches so that there are 30 leaf nodes. As a result, some leaves in the plot correspond to more than one data point.

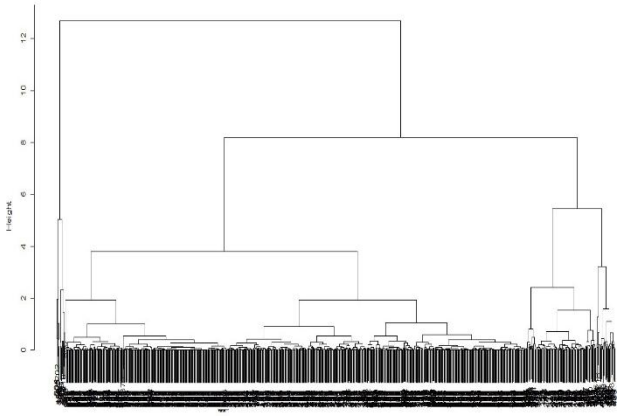


Figure 6. Hierarchical cluster binary tree of data points

Conclusion

Therefore, in this project, we here by managed to find the air pollution using NODE MCU and the air sensor. It would be really helpful in situations like gas leak in the households and even on the industries. We have also set a margin on

the data so once it reaches a certain amount the air is detected and we can immediately take appropriate action. for giving me the courage and the strength that I needed to complete my goal. This acknowledgment would be incomplete without expressing the whole hearted thanks to my family and friends who motivated me during the course of my work.

Acknowledgments

I would like to thank Director of MCA Department Dr.P. Radha, and Mr. A.D.C. Navin Dhinnesh, Assistant Professor (Sr. Gr), Department of Computer Applications for their motivation and support throughout the project.

References

1. Smart air pollution monitoring system by Okokpujie, Kennedy O. and Noma-Osaghae, Etinosa and Odusami [5]
2. https://ieeexplore.ieee.org/abstract/document/8663367/
3. 3. S. Kumar and A. Jasuja. Air quality monitoring system based on internet of things using raspberry pi. Pages 1341-1346, May 2017.
4. Navin Dhinnesh, A.D.C., Sabapathi, T. Probabilistic neural network based efficient bandwidth allocation in wireless sensor networks. *J Ambient Intell Human Comput* **13,** 2001–2012 (2022).
5. 5 Navin Dhinnesh, A.D.C., Sabapathi, T, and Sundareswaran N, An efficient self-healing network through quadratic probing mechanism, Journal of communication systems, 35, (7), 2022
6. Neethidevan. V, "Facial Recognition in Web Camera using Deep Learning under Google COLAB" GRD Journals-Global Research and Development Journal for Engineering vol5. Issue 7,pp.53-57, June-2020,
7. Neethidevan. V , "Image Classification using Pre-Trained Convolutional Neural Network in COLAB" GRD Journals-Global Research and Development Journal for Engineering vol5. Issue 7,pp.40-45, June-2020,
8. Navin Dhinnesh ADC, "Wireless Sensor Networks and its Tools for Simulation" GRD Journals-Global Research and Development Journal for Engineering vol. 4. Issue 10, pp. 45 - 48, 2019,
9. Developing of an IoT-Based indoor air quality monitoring platform by JunHo Jo [3] 14
10. S.R. Enigella and H. Shahnasser. Real-time air quality monitoring. Pages 182-185, Jan 2018
11. 11. Sumanth Reddy Enigella, Hamid Shahnasser. "Real-Time Air Quality Monitoring", 2018 10th International Conference on Knowledge and Smart Technology (KST), 2018.
12. https://www.kaggle.com/datasets
13. https://www.datanovia.com/en/blog/k-means-clustering-visualization-in-r-step-by-stepguide/
14. https://stanford.edu/class/engr108/visualizations/kmeans/kmeans.html#:~:text=Mean%20square%20point%2Dcentroid%20distance,groups%20or%20clusters%20of%20points.