

# Improve air quality tracking through a fully intelligent IoT-based program with AERO Virtual Assistant.

Venkata Rao Yanamadni<sup>1\*</sup>, T. Shesagiri<sup>2</sup>, Palamakula Swathi<sup>2</sup>, Pattipati Naveen Kumar<sup>2</sup>, Harikrishna Bommala<sup>1</sup>, G. vijendar reddy<sup>3</sup>, Kseniia Iurevna, Usanova<sup>4</sup>

<sup>1</sup>Department of CSE, KG Reddy College of Engineering & Technology, Moinabad, Hyderabad, Telangana, India.

<sup>2</sup>Department of CSE Joginapally B.R. Engineering College, Yenkapally, Hyderabad, Telangana, India

<sup>3</sup>Department of IT, GRIET, Hyderabad, Telangana, India

<sup>4</sup>Lovely Professional University, Phagwara, Punjab, India.

**Abstract.** Have you thought about the quality of the air we breathe? Living a healthy and happy life is so important in the wake of the COVID-19 pandemic. To live a healthy life, everyone needs to be aware of the quality of the air they breathe and how their actions impact current and future air quality. This project presents his IoT-based system for measuring, monitoring, communicating, and predicting future air quality. The system was developed using a combination of IoT and machine learning to monitor, analyze and predict air quality in real-time. Air quality can be measured using various gas sensors, temperature and humidity sensors. It also measures the number of hazardous gases emitted in industrial areas. Machine learning models are applied to data values from various sensors to predict future air quality. The proposed solution monitors the current air quality on the device through a small smart application, observes fluctuations using graphs, and sends SMS alerts if air quality parameters exceed acceptable values. Another important feature is the addition of AERO, a virtual assistant that allows anyone to communicate atmospheric details. This system allows businesses, organizations, industries, farmers, gardeners, healthcare workers, city dwellers, etc. to easily connect, communicate, and monitor the current air quality in their daily lives and lead healthier lives. You will be able to. This will allow you to perform the necessary actions.

## 1 Introduction

Air quality is currently a growing concern around the world. Due to increasing industrialization and urbanization, air pollution poses a serious threat to both the environment and human health, with air pollution claiming approximately 8 million lives each year. Air pollution not only affects mortality rates but also causes various health problems such as lung diseases, cardiovascular diseases, skin cancer, and other serious effects [1]. Additionally, air

---

\* Corresponding author: [venkataraoyanamadni@gmail.com](mailto:venkataraoyanamadni@gmail.com)

pollution increases the concentration of harmful pollutants such as ammonia, carbon monoxide, methane, hydrogen, nitrogen, and phosphorus, leading to climate change, acid rain, ozone depletion, and biodiversity loss. All living things breathe in and breathe out air containing carbon dioxide and oxygen. Therefore, we need to maintain and monitor air quality [1]. The air present consists of several harmful gases that are harmful to the environment and public health [3].

In this context, the Internet of Things (IoT) has been recognized as a promising technology for real-time monitoring of air quality, as presented in [2]. The proposed solution is a step forward in being able to measure, monitor, communicate and predict future air quality. This model includes various sensors such as MQ135 (measuring CO<sub>2</sub>, NO<sub>x</sub>, SO<sub>2</sub>, NH<sub>3</sub>, and benzene) and DHT11 (high sensitivity sensor measuring humidity and temperature) [10]. It keeps us safe by monitoring the toxic gases present in our surroundings, comparing them to acceptable levels, and sending SMS messages to users' mobile devices if they exceed a certain value [4]. Masu. Additionally, the device covers a multi-criteria system for predicting air quality based on time series forecasts. Another unique thing about this built-in system is the addition of a virtual assistant called AERO (a Greek word that translates to AIR). This virtual assistant allows all users to: B. For city dwellers, gardeners, farmers, etc. to easily communicate details of the current atmosphere.

## 2 Literature review

this reference paper, “IoT-based Air Pollution Monitoring and Prediction System,” the authors focused on minimizing air pollution and its associated health impacts for urban residents. Connecting the four sensors MQ-135, MQ-7, Sharp Dust Sensor and DHT11 to the Arduino Mega 2560 microcontroller, Arduino has its own IDE (Integrated Development Environment) and C/C++ etc. Equipped with programming language. The authors wrote the necessary code in the Arduino IDE and uploaded it to the Arduino board to retrieve sensor data. Real-time cloud storage ThingSpeak is used to read sensor data using the read API. The acquired data is preprocessed. The ARIMA model was used to predict future pollutants.

In this reference paper, “IoT Based Sound and Air Pollution Measuring Temperature and Humidity Monitoring System”, the authors have introduced a single IoT system that combines a monitoring and alerting system for air, noise pollution, temperature, and humidity. A mobile application that was developed by using IoT technology is used to monitor the values of the gas sensors, soil moisture sensors, and sound sensors that are coupled to the Arduino UNO [3].

In this reference paper, “Air Quality Monitoring Systems using IoT: A Review”, the authors have created an Air Quality System with nodes carrying inexpensive sensors installed in buildings, and the monitoring is made available at a server through IoT. These monitoring nodes were deployed in fixed locations and provided information on spatiotemporal variation at various sites, pollution sites, and sources of pollution. For real-time monitoring of air quality, MOS (Metal Oxide Semiconductors) sensors were utilized, however, they demand high temperatures for quick reaction times and are less sensitive to changes in humidity and temperature. MOS sensors are cross sensitive and have a nonlinear response curve [7].

In this reference paper, “Design and Analysis of IoT based Air Quality Monitoring System”, the authors have proposed an IoT device for monitoring air quality using an MQ9 gas sensor (sensitive to measure CO), MQ2 sensor (to detect smoke) and PMS3003 G3 particle sensor which measures PM<sub>2.5</sub> in real-time. These readings are given to Node MCU,

which immediately transmits the detected data to ThingSpeak through the internet. Hyper Text Transfer Protocol (HTTP) is utilized to access the data stored in Thingspeak over the internet. This platform allows for the graphical plotting of sensed data for better analysis, which may then be seen on an OLED display [9].

### 3 Implementation of The System

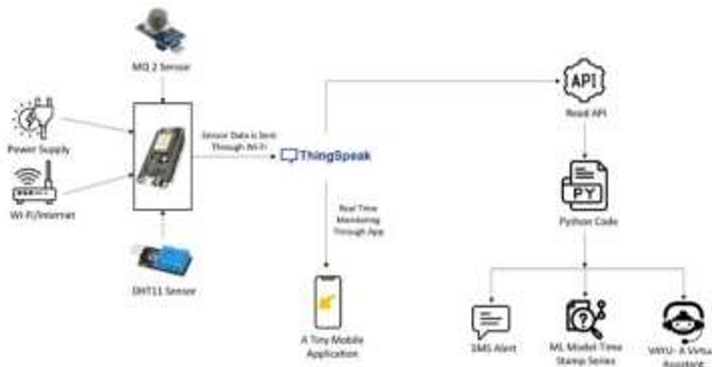
In Fig. 1, a system architecture is used to depict the system's overview. Two sensors—the MQ2 and DHT11 sensors - as well as a Microcontroller - NodeMCU is connected to develop the system. Through the Wi-Fi Module, sensor data is transmitted to an open-source real-time cloud storage system (Thingspeak). The fetched data is used for data visualization and future analysis.

#### 3.1 System Architecture

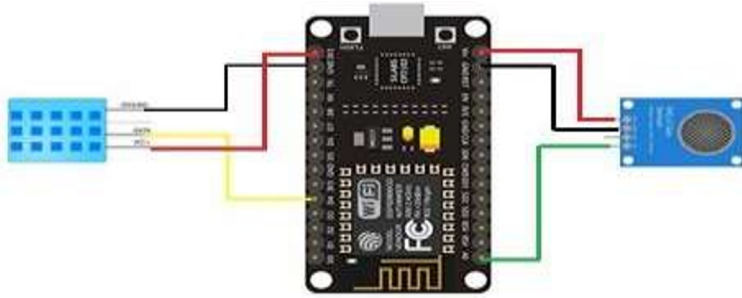
Dual sensors MQ2 and DHT11 sensors are connected to the microcontroller NodeMCU. MQ2 sensor is used to measure air pollutants such as CO, Methane, Propane, Hydrogen, Smoke, and LPG. To measure the levels of Temperature and Humidity, a DHT11 sensor is used. In this setup Fig. 2, the analog output pin of the MQ2 sensor is connected to the A0 analog input pin of NodeMCU. The MQ2 sensor provides an analog output that varies based on the concentration of the pollutants being detected. On the other hand, the DHT11 sensor uses a digital signal protocol to communicate temperature and humidity data. One of the digital input pins (D0 to D8) of NodeMCU should be connected to the data pin of the DHT11 sensor. The NodeMCU board is based on the ESP8266 microcontroller which can be programmed using Arduino-based languages.

In Arduino IDE, we wrote the necessary code to fetch the sensor data and also set up the Wi-Fi module on the NodeMCU board by providing the SSID and password of the Wi-Fi network that you want to connect to. After setting up, use the Thingspeak API to send sensor data to Thingspeak.

Once the data is sent, we can use the read API of Thingspeak to retrieve the data for preprocessing. For forecasting future air quality, the model that is suitable for our dataset is ARIMA Model. The preprocessed data is used to train our model to predict future air quality by considering all the parameters.



**Fig. 1.** System Architecture



**Fig. 2.** Hardware Setup

### 3.2 Hardware and Software Requirements

The hardware equipment and software implementation required for the development of the proposed model are:

**NodeMCU:** It is an open-source firmware and development kit that makes it simple to build Wi-Fi-enabled IoT (Internet of Things) applications. It is built around the ESP8266 microcontroller, which is a low-cost, low-power, and highly-integrated chip that provides Wi-Fi connectivity. It offers a range of integrated modules and libraries that facilitate the interaction with a range of sensors, actuators, and other IoT devices. Additionally, NodeMCU is highly adaptable and extendable, enabling us to modify the firmware to meet our unique requirements.

**MQ-2 Sensor:** The MQ-2 sensor analyses the amount of pollutants in the air, such as carbon monoxide, methane, propane, hydrogen, smoke, and other pollutants. The MQ2 sensor is excellent for a variety of applications and has a high sensitivity and detects a wide range of gases. The MQ2 sensor works on the principle of metal oxide semiconductors, where the presence of certain gases causes a change in the sensor's resistance. The MQ2 sensor module has four pins, including VCC, GND, analog output, and digital output. The analog output provides a voltage output that varies with the concentration of the detected gas, while the digital output provides a binary signal that indicates the presence or absence of gas above a preset threshold level.

**DHT11 Sensor.** It is used to measure environmental conditions such as temperature and humidity. It has a range of 0-50 degrees Celsius for temperature measurements with an accuracy of  $\pm 2$  degrees Celsius and a range of 20-80% for humidity measurements with an accuracy of  $\pm 5\%$ . It consists of four pins: VCC, GND, data, and NC (Not Connected). It operates on a voltage range of 3.3V to 5V and draws less than 1mA of current during active measurements.

**Arduino IDE:** The Arduino IDE is a software platform for developing and configuring Arduino boards. It provides a simple and easy-to-use interface that allows users to create, build, and upload code to Arduino boards. It is based on the Processing language and is compatible with Windows, macOS, and Linux operating systems. It provides a code editor with syntax highlighting, auto-completion, and error - highlighting features that make it easy to write and debug code. Additionally, it has a serial monitor that enables users to interact with their Arduino board and monitor the output of their programs. The Arduino IDE provides a library manager that makes it simple for users to add and manage libraries, which are pre-written code modules that can be used to add extra functionality to projects. It also

supports a wide range of third-party libraries that can be downloaded and installed from various sources. It supports a wide range of sensors, actuators, and other hardware components that can be easily interfaced with Arduino boards.

**ThingsSpeak:** It is a free, open source, real-time, cloud- based IoT platform that enables users to gather, examine, display and visualize data from IoT devices. The platform provides APIs and other tools for data analysis and visualization, which allow users to monitor and comprehend their data in real time. ThingsSpeak also provides a range of tools for data visualization, including graphs, charts, and maps, so customers can quickly visualize their data and gain insights about their IoT systems. Additionally, ThingsSpeak provides support for integrating with other popular IoT platforms, such as IFTTT (If This Then That), which enables users to automate actions based on their IoT data.

**VS Code:** It is a free and open-source text editor that runs on Linux, macOS, and Windows. The editor is low weight and robust, it has several powerful features that helped VS Code to become one of the most popular development environment tools in recent years. It supports a large number of programming languages including Java, Python, C++, C#, HTML, CSS, and more. It also provides extensions to add new themes, debuggers, commands, and support for web programming.

### 3.3 ML Model

A machine learning algorithm can be implemented for better understanding and prediction of future air pollutants. The algorithm intakes the sensor values and uses those environmental parameters to forecast the pollutant levels more accurately. We have selected a model for time series analysis of future air quality: ARIMA model. The usage of time series models is to understand underlying forces and structure of observed data and proceed to forecasting, monitoring or feed forward control. The accuracy of the model can be measured in terms of MSE, RMSE, etc.

#### 3.3.1 ARIMA ML Model

Auto Regressive Integrated Moving Average is referred to as ARIMA. This method of time series analysis predicts a series of future values based on its historical values. It is a statistical method of serial correlation, in which past data is used to influence future data points. It is composed of 3 elements: ‘AR’ refers to autoregression, is a model in which the future values of the time series can be predicted based on its own past behavior. ‘I’ stands for Integration, is a model used in most of the statistical model to transform the non- stationary time series into a stationary making it suitable for modeling and prediction. ‘MA’ known as moving average, is model that captures the randomness or noise in time series. Thus, the future values can be predicted based on the error from previous forecast, resulting in a more accurate prediction.

The entire model is signified as ARIMA(a, b, c), where ‘a’ is the number of past data points that will be used to predict the future value of the time series, ‘c’ is the number of past forecast errors used in the prediction equation, and ‘b’ represents the number of times the time series needs to be differenced to achieve stationarity, also known as the degree of differencing. Auto Regression is a linear regression model that employs its own lags as predictors. A pure Auto Regression (AR) model can be represented by an equation:

$$X(t) = \text{constant} + \phi(1)X(t-1) + \dots + \phi(p)X(t-p) + \epsilon(t) \quad (1)$$

That is,  $X(t)$  is a function of the ‘lags of  $X(t)$ ’.

A pure Moving Average is one in which  $X(t)$  depends only on lagged forecast errors. The equation can be written as:

$$X(t) = \text{constant} + \theta(1)e(t-1) + \dots + \theta(q)e(t-q) + e(t) \quad (2)$$

Here,  $X(t)$  is time series value at time  $t$ ,  $c$  is constant,  $\phi(1)$ ,  $\phi(p)$  are autoregressive coefficients,  $\theta(1)$ ,  $\dots$ ,  $\theta(q)$  are moving average coefficients, and  $e(t)$  is the error at time  $t$ . The model can be represented by an equation:

$$X(t) = \text{constant} + \phi(1)X(t-1) + \dots + \phi(p)X(t-p) + \theta(1)e(t-1) + \dots + \theta(q)e(t-q) + e(t) \quad (3)$$

In words, Prediction = constant + linear combination lags of  $X$  + linear combination of lagged predicted errors.

### 3.3.2 Measures of Accuracy

Mean Absolute Error: The arithmetic mean of the absolute difference between the model forecast and desired value.

$$\text{MAE} = (1/n) * \sum |X_i - Y_i| \quad (4)$$

Mean Squared Error: It measures the amount of inaccuracy in statistical models. It evaluates the arithmetical mean squared deviation between observed values and predictions.

$$\text{MSE} = (1/n) * \sum (X_i - Y_i)^2 \quad (5)$$

Root Mean Squared Error: It determines the square root of the mean squared error (MSE).

$$\text{RMSE} = \text{sqrt}(1/n) * \sum (X_i - Y_i)^2 \quad (6)$$

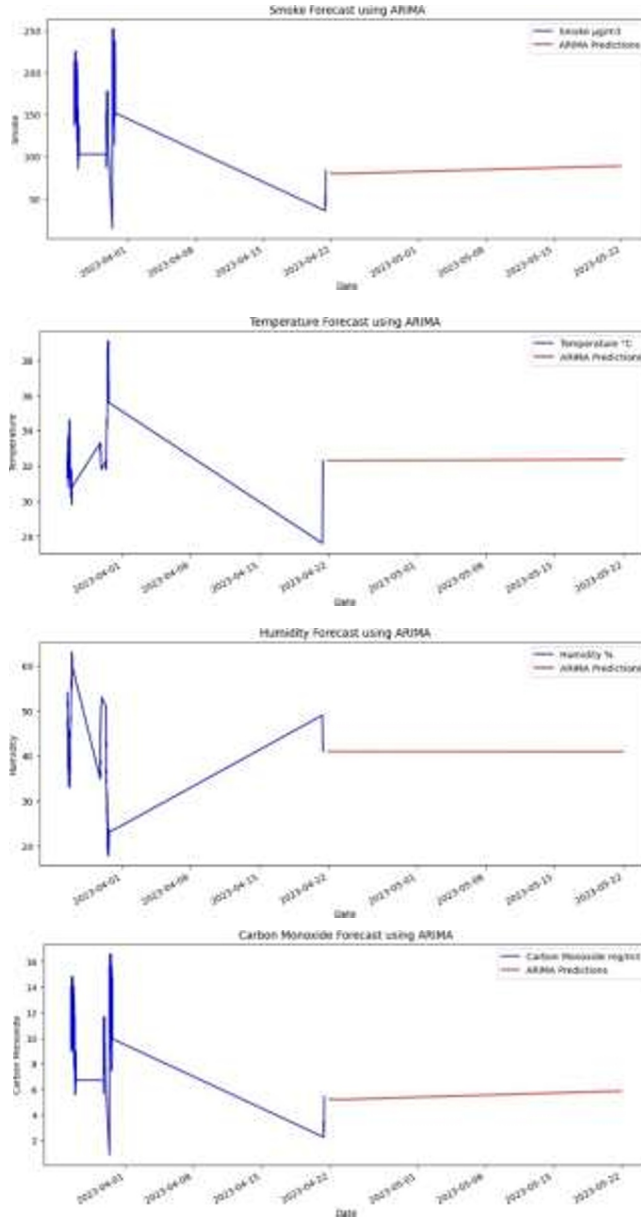
Mean Absolute Percentage Error: This is the mean absolute error's (MAE) percentage equivalent. It measures the average magnitude of the model's error, or how typically inaccurate forecasts are.

$$\text{MAPE} = (1/n) * \sum (|X_i - Y_i| / X_i) * 100 \quad (7)$$

In which,  $X_i$  is the original value,  $Y_i$  is the forecast value, and  $n$  is total observations.

These measurements may be used to evaluate the accuracy of an ARIMA model and analyze the performance of various models. A model with lower values of these measures is considered to be more accurate and realistic.

Fig. 3 illustrates the values of both the original and forecast data of Temperature, Humidity, CO, Methane, Propane, Butane, Hydrogen, Smoke, and LPG of our system. X-axis appears to be the date and Y-axis appears to be the pollutant parameters.



**Fig. 3.** Actual Vs ARIMA Predictions

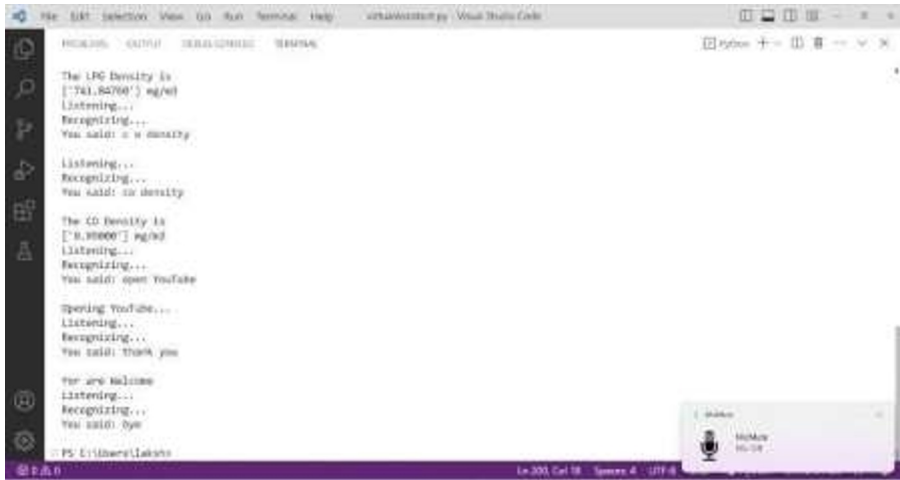
### 3.4 Virtual Assistant – Aero

“Aero” is a virtual desktop assistant developed to assist the user in understanding the system and system function. This feature allows our machine to run according to our commands and respond to user queries. Fig 4 depicts the response of “Aero” to the user queries. Python provides many built-in modules for making a virtual assistant.

1. pytsx3: pytsx3 is a text-to-speech library used for text-to- speech conversion. It is platform-independent and works offline. It is used to generate speech from written text in

a range of varied voices and languages. It supports multiple TTS engines like Microsoft Speech API (SAPI), eSpeak, and Google Text-to-Speech(gTTS).

2. **SpeechRecognition:** It is a Python module for converting speech to text. The module supports several audio file formats including WAV, AIFF, FLAC, and MP3, also streaming audio from microphones.
3. **Wikipedia:** It is a Python module for retrieving information from Wikipedia websites. This makes our device more interactive and communicable. It is a useful tool for accessing information from the world's largest online encyclopedia and is used in many Python applications.
4. **webbrowser:** A Python module to interact with web browsers. It allows users to launch a browser and open URLs. For example, the virtual assistant is capable of opening Google and YouTube by their URLs.



**Fig. 4.** Aero – Virtual Assistant

### 3.4.1 Aero Application

“Aero” is a basic application designed to monitor real time trend of air quality. It is developed using MIT App Inventor that allows users to create mobile applications for Android devices. It uses a visual block-based programming language, where users can drag and drop blocks of code to create their custom apps. There are wide range of blocks for different components such as buttons, text boxes, images, websites etc. It provides interactive user interface design. To install the app on our Android device, we can scan the QR code and install it in our device through APK.

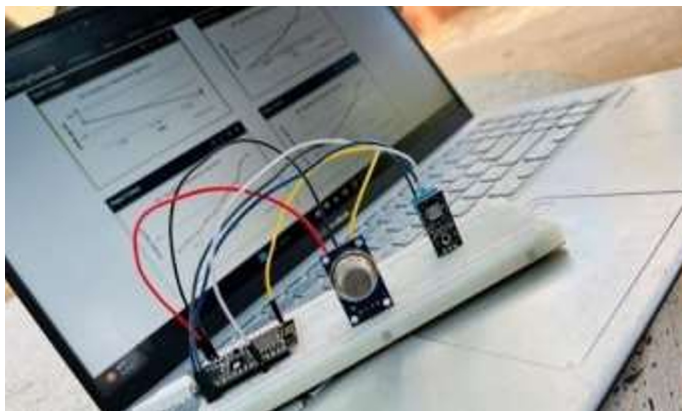


**Fig. 5.** Aero Application

### 3.5 Visualization and Analysis

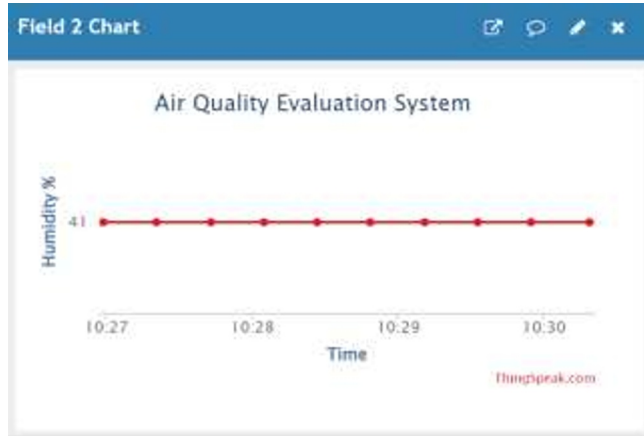
#### 3.5.1 Visualization

The system in which sensors connected to NodeMCU is placed at a location to evaluate the air contaminants. The raise in the permissible levels of the pollutants causes an SMS alert which indicates that the necessary action should be taken. The sensor data was gathered for every 5 minutes which is represented through charts for every pollutant differently using ThingSpeak.



**Fig. 6.** Device Setup

For future analysis, the ARIMA model suited well for our dataset, in which the entire data was split into Train and Test sections. The model deal with noise and outliers in a better way and future predictions were successful. We have considered different metrics such as MAE, RMSE, MAPE and MSE to determine the accuracy of the model. We were able to get lower MAE, RMSE, and MAPE values, which indicates superior model performance.



**Fig. 7.** Typical Humidity Chart



**Fig. 8.** Typical CO Chart

A tiny application “Aero” was capable of demonstrating air quality in real time and empowering every individual for better understanding of human impact on environment. Also, Virtual Assistant “Aero” was able to respond to the query quickly and made communication easier and it took average of 3 secs to get activated.

## 4 Results

We have successfully developed an IoT device that is capable of measuring, monitoring, communicating and forecasting the air quality. The Virtual Assistant was integrated with ARIMA model to assist every individual such as urbanites, farmers, gardeners etc., for better understanding of the air quality and helped in predicting the future trend of the sensors. It was integrated with Wikipedia, Google Search Engine to offer more details about the air quality, weather updates, and many other topics. A real time monitoring through “Aero” application was successfully installed in mobile for better analysis.

## 5 Future Scope

The system has a wide scope to make it more advanced and interactive. It can be expanded to cover more areas to provide a wider range of air quality data and personalized health advisories to individuals based on their location, health status and the current air quality. This can help people to take proactive measures to protect their health as well as environment.

## 6 Conclusion

We have successfully calculated various poisonous gases such as Carbon Monoxide, Methane, Hydrogen, Propane, Smoke, LPG and environmental parameters such as Temperature and Humidity. We have developed a system through which a user can monitor, measure, analyze and communicate from anywhere by sending sensor data to cloud via NodeMCU (Wi-Fi Module). To analyze present data and forecast future data we have implemented ARIMA model to train our dataset and predict the level of pollutants in the future.

Thus, this system is suitable for achieving healthy environment. By which, measures should be undertaken and raise awareness in each and every individual. The model can be integrated with improved sensors and make it more communicable and easier to use.

## References

1. Mohammed Rakib, Sanaula Haq, Md. Ismail Hossain and Tanzilur Rahman, "IoT Based Air Pollution Monitoring & Prediction System", ICISSET, 2022.
2. Danny Munera, Diana P. Tobon V, Johnny Aguirre and Natalia Gaviria Gomez, "IoT-based air quality monitoring systems for smart cities: A systematic mapping study", International Journal of Electrical and Computer Engineering (IJECE), 2021.
3. P. V. Hari Prasad, Maddasani Daya Rani, Garimella Keerthana, Kollu Yaswanth Kumar and Uppalapathi Bhargav, "IoT Based Sound and Air Pollution Measuring Temperature and Humidity Monitoring System", International Journal of Innovative Research in Computer Science & Technology (IJIRCST), 2021.
4. Agiru Hima Vasanth, Ayush Agrawal, Aditya Bohara, Jagjeet Beberta, Dr. Muzameel Ahmed and Dr. Suma V, "Air Pollution Monitoring System Using IOT", International Journal of Creative Research Thoughts (IJCRT), 2021.
5. Anand Jayakumar A, Praviss Yesyand T K, Venkatesh Prashanth K K and Ramkumar K, "IoT Based Air Pollution Monitoring System", International Research Journal of Engineering and Technology (IRJET), 2021.
6. Helton Pierre Lucena de Medeiros and Gustavo Girao, "An IoT-based Air Quality Monitoring Platform", 2020.
7. Virendra Barot and Viral Kapadia, "Air Quality Monitoring Systems using IoT: A Review", International Conference on Computational Performance Evaluation (ComPE), 2020.
8. R Santhiya, D Saritha, E Vaijyanthi, S Vishalini and Dr. K Kalaiselvi, "IoT Based Air Defilement Observing System", International Research Journal of Engineering and Technology (IRJET), 2020,

9. Ajitesh Kumar, Mona Kumari and Harsh Gupta, “Design and Analysis of IoT based Air Quality Monitoring System”, International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), 2020.
10. T. Veeramankandasamy, Gokul Raj.S, A. Balamurugan, A.P. Ramesh and Y.A. Syed Khadar, “IoT based Real-time Air Quality Monitoring and Control System to Improve the Health and Safety of Industrial Workers”, International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2020.
11. Martin M. Soto-Cordova, Martha Medina-De-La-Cruz and Anderson Mujaico-Mariano, “An IoT based Urban Areas Air Quality Monitoring Prototype”, International Journal of Advanced Computer Science and Applications (IJACSA), 2020.
12. Attapon Udomlumlert, Napaphat Vichaidis, Pichitchai Kamin, Thattapon Surasak, Scott C.-H. Huang and Nattagit Jiteurtragool, “The Development of Air Quality Monitoring System using IoT and LPWAN”, 2020.
13. Cynthia J, Saroja M.N, Parveen Sultana and J. Senthil, “IoT-Based Real Time Air Pollution Monitoring System”, International Journal of Grid and High Performance Computing, 2019.
14. Ramik Rawal, “Air Quality Monitoring System”, International Journal of Computational Science and Engineering, 2019.
15. Vijayakumar Sajjan and Pramod Sharma, “Research on an IoT Based Air Pollution Monitoring System”, International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019.
16. Harsh Gupta, Dhananjay Bhardwaj, Himanshu Agrawal, Vinay Anand Tikkiwal and Arun Kumar, “An IoT Based Air Pollution Monitoring System for Smart Cities”, ICSETS, 2019.
17. Ayaskanta Mishra, “Air Pollution Monitoring System based on IoT: Forecasting and Predictive Modeling using Machine Learning”, International Conference on Applied Electromagnetics, Signal Processing and Communication (AESPC), 2018.
18. Kennedy Okokpujie, Etinosa Noma-Osaghae, Odusami Modupe, Samuel John and Oluga Oluwatosin, “A Smart Air Pollution Monitoring System”, International Journal of Civil Engineering and Technology, 2018.
19. Harsh N. Shah, Zishan Khan, Abbas Ali Merchant, Moin Moghal, Aamir Shaikh and Priti Rane, “IOT Based Air Pollution Monitoring System”, International Journal of Scientific & Engineering Research, 2018.
20. Pattar Sunil Mahesh, Patil Bhushan Rajendra, Bodke Akshay Dnyaneshwar and Mr. Ulhas. A. Patil, “A Survey Paper on Air Pollution Monitoring Using IoT”, IJARIE, 2018.
21. Temesegan Walelign Ayele and Rutvik Mehta, “Air pollution monitoring and prediction using IoT”, d International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018.
22. Gagan Parmar, Sagar Lakhani and Manju K. Chattopadhyay, “An IoT Based Low Cost Air Pollution Monitoring System”, International conference on Recent Innovations in Signal Processing and Embedded Systems (RISE), 2017.
23. Raghavendra Khot and Prof. Vidya Chitre, “Survey on Air Pollution Monitoring Systems”, International Conference on Innovation in Information, Embedded and Communication Systems (ICIIECS), 2017.