International Journal of Scientific Research & Engineering Trends
Volume 7, Issue 4, July-Aug-2020, ISSN (Online): 2395-566X

IoT Based Air Pollution Monitoring System

Ashwini, Anvitha Poojary, Anvitha U, Akshata Hedge, Prof. Dr. Mohideen Badhusha S
Department of Computer Science and Engineering,
Alva’s Institute of Engineering & Technology, Moodbidri
badhusha.sm@gmail.com

***Abstract*- Internet of Things (IoT) is a worldwide system of “smart devices” that can sense and connect with their surroundings and interact with users and other systems. Global air pollution is one of the major concerns of our era. Existing monitoring systems have inferior precision, low sensitivity, and require laboratory analysis. Therefore, improved monitoring systems are needed. To overcome the problems of existing systems, we propose a three-phase air pollution monitoring system. An IoT kit was prepared using gas sensors, Arduino IDE (Integrated Development Environment), and a Wi-Fi module. This kit can be physically placed in various cities to monitoring air pollution. The gas sensors gather data from air and forward the data to the Arduino IDE. The Arduino IDE transmits the data to the cloud via the Wi-Fi module. We also developed a Web front-end so that users can view and access relevant air quality data from the cloud.**

***Keywords*- Smart devices, Wi-Fi module etc.**

I.INTRODUCTION

Wireless Sensor Network is progressively affecting everyday living. A WSN is a network consisting of sensor nodes. Each sensor can detect certain factors like air pressure, air composition, and water quality.WSNs are used in a wide variety of settings, including personal space, industrial floors, agriculture, home utility monitoring systems, factory automation, automotive, and many other fields. WSNs are related to the concept of IoT.In IoT, devices are interconnected to transmit data via distributed sensor networks. IOT has useful applications in the medical field.

Devices such as smartphones and sensing systems can be associated to create an infrastructure that provides access to health care information and services.This approach is referred to as “Mobile-Health”. Mobile-Health can be viewed as the consequence of the convergence of wireless communication systems, WSNs, andglobal computing tools. As per WHO estimates, 10 out of 20 most populated cities in the world are in India. Based on the concentrations of PM2.5 emissions of

India was ranked fifth most polluted country by WHO in which 21 among top 30 polluted cities were in India. Around 90 % of the population in low and center wage nations is presented to perilous levels of encompassing air contamination. The World Bank works with creating nations and advancement accomplices to diminish contamination by supporting checking and examination, administrative changes, and ventures. In 2016 for instance, the Bank conferred US$1 billion to enable China to enhance air quality by lessening discharges of particular air poisons from mechanical, transport and country sources in the territory of Hebei, and by expanding vitality effectiveness and clean vitality through imaginative financing in the Beijing-Tianjin-Hebei district (otherwise called Jing-Jin-Ji locale) that covers the capital region and neighboring regions.

Death associated with encompassing air contamination have expanded in intensely populated, quick urbanizing areas, while death identified with cooking and warming homes with strong energizes have stayedconsistent in spite of advancement additions and upgrades in wellbeing administrations. Illnesses credited to the two sorts of air contamination caused 1 of every 10 deaths in, at least 2013 than six times the quantity of death caused by intestinal sickness.

1. The Cost of Air Pollution:
Strengthening the financial case for activity, a joint investigation of the World Bank and the Institute for Health Metrics and Evaluation (IHME), looks to appraise the expenses of unexpected losses identified with air contamination, to fortify the case for activity and encourage basic leadership with regards to rare assets.

An evaluated 5.5 million lives were lost in 2013 to illnesses related with outside and family unit air contamination, causing human enduring and decreasing financial improvement.In this paper, we describe the implementation of an IoT mobile-air pollution detection device that detects the quantity of any gas in the air and comparesit with threshold, if the range is above the threshold then a message is triggered and the user is notified.

II. PROBLEM STATEMENT

Pollution can build up in isolated pockets, and local sources (an industrial plant or a busy road) can add to the overall poor air quality. A network of citizen scientists monitoring air quality throughout a region could help reveal how pollution travels through the region and could help identify pollution "hot spots."

There are many cities around the world facing air quality issues. The contaminated air results in death every year and decline in health conditions as people are exposed to unhealthy air quality. Awareness of the contaminated air enables the community to take precautionary steps. This will also enable the relevant authority to take remedial action. With this project the community can enjoy cleaner air and improved health conditions.

III. PROPOSED SYSTEM

The proposed system is designed with following modules:

- Switch On
- Reading the gas levels
- Plot and check the levels
- Trigger the function

1. Device Functions:
once the device is turned on the sensor reads the level of the gas that it has been designed for. The device can be modified according to different gases by changing the IC's or integrating them. The device is designed such it reads gas level in air every 3 seconds. Each reading is plotted on a graph in the web front-end. After a required time limit all reading are added and derive a mean out of it. The readings along with the mean are updated in the database each time there is a new reading.

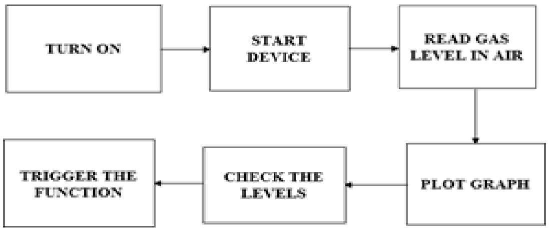


Fig 1. System Architecture.

2. Comparing the levels:
Once the readings are uploaded onto database the mean is compared with threshold. If mean is greater that threshold it trigger Lambda function from then AWS services. Lambda contains a written function that triggers Twilio to send SMS to the user once the levels of any harmful gas are higher that threshold.

3. Triggering the SMS:
AWS Lambda allows us to write functions that can run on specific conditions being attained. We use this feature of Lambda totrigger Twilio SMS whenthe mean gas level exceed the threshold levels.

IV. LITERATURE SURVEY

G. Lo Re, D. Peri, and S. D. Vassallo, "Urban air quality monitoring using vehicular sensor networks," in Advances onto the Internet of Things, Springer
Air pollution in large urban areas has a drastic effect on humans and the environment. Ecological issues in India are growing quickly. Air contamination is mainly caused by vehicles and industries which cause various respiratory diseases such as asthma and sinusitis.

The quality of air is inferior in metropolitan cities like Kolkata, Delhi, and Mumbai due to a large amount of carbon dioxide and other harmful gases emitted from vehicles and industries. An extensive number of projects have been described in the literature that utilize low-cost air pollution sensing devices that can be carried by individuals or by versatile vehicles.

J. J. Caubel, T. E. Cados, and T. W. Kirchstetter, "A New Black Carbon Sensor for Dense Air Quality Monitoring Networks," Sensors, vol. 18, no. 3, p. 738, 2018.
This paper is highlighting the outliers about Low-cost air pollution sensors are emerging and increasingly being deployed in densely distributed wireless networks that provide more spatial resolution than is typical in traditional monitoring of ambient air quality. However, a low-cost option to measure black carbon (BC)—a majorcomponent of particulate matter pollution associated with adverse human health risks—is missing.

This paper presents a new BC sensor designed to fill this gap, the Aerosol Black Carbon Detector (ABCD), which incorporates a compact weatherproofenclosure, solar-powered rechargeable battery, and cellular communication to enable long-term, remote operation.

This paper also demonstrates a data processing methodology that reduces the ABCD'ssensitivity to ambient temperature fluctuations, and therefore improves measurement performance in unconditioned operating environments (e.g., outdoors). A fleet of over 100 ABCDs was operated outdoors in collocation with a commercial BC instrument (Magee Scientific, Model AE33) housed inside a regulatory air quality monitoring station.

The measurement performance of the 105 ABCDs is comparable to the AE33. The fleet-average precision and accuracy, expressed in terms of mean absolute percentage error, are 9.2 ± 0.8% (relative to the fleet average data) and 24.6 ± 0.9% (relative to the AE33 data), respectively (fleet-average ± 90% confidence interval).

L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, "Field calibration of a cluster of low-cost commercially available sensors for air quality

monitoring. Part B: NO, CO and CO2," Sensors Actuators B Chem., vol. 238, pp. 706–715, 2017
In this work the performances of several field calibration methods for low-cost sensors, including linear/multi linear regression and supervised learning techniques, are compared. A cluster of either metal oxide or electrochemical sensors for nitrogen monoxide and carbon monoxide together with miniaturized infra-red carbon dioxide sensors was operated. Calibration was carried out during the two first weeks of evaluation against reference measurements.

The accuracy of each regression method was evaluated on a five months field experiment at a semi-rural site using different indicators and techniques: orthogonal regression, target diagram, measurement uncertainty and drifts over time of sensor predictions. In addition to the analyses for ozone and nitrogen oxide already published in Part a [1], this work assessed if carbon monoxide sensors can reach the Data Quality Objective (DQOs) of 25% of uncertainty set in the European Air Quality Directive for indicative methods.

As for ozone and nitrogen oxide, it was found for NO, CO and CO2 that the best agreement between sensors and referencemeasurements was observed for supervised learning techniques compared to linear and multilinear regression.

Sharada P. Mohanty, David P. Hughesand Marcel Salathé "Using Deep Learning for Image-Based Plant Disease Detection."
Crop diseases are a major threat to food security, but their rapid identificationremains difficult in many parts of the world due to the lack of the necessary infrastructure.

The combination of increasing global smartphone penetration and recent advances in computer vision made possible by deep learning has paved the way for smartphone-assisted disease diagnosis. Using a public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions, we train a deep convolutional neural network to identify 14 crop species and 26 diseases (or absence thereof).

The trained model achieves an accuracy of 99.35% on a held- out test set, demonstrating the feasibility of this approach. Overall, the approach of training deep learning models on increasingly large and publicly available image datasets presents a clear path toward smartphone-assisted crop disease diagnosis on a massive global scale.

S. Gaglio, G. Lo Re, G. Martorella, D. Peri, and S. D. Vassallo, "Development of an IoT environmental monitoring application with a novel middleware for resource constrained devices," in Proceedings of the 2nd Conference on Mobile and Information Technologies in Medicine (MobileMed 2014), 2014

In this paper the development of a Mobile Health monitoring system is described. The system combines user location data with air quality information provided by a heterogeneous sensing infrastructure providing users with advises about their daily exposure to air pollutants.

The highly dynamic integration of different kind of nodes, mostly characterized by rather constrained resources, of this application is crucial to implement the Internet of Things vision, and requirespowerful and effective programming methodologies to abstract implementation of high-level distributed processing from hardware dependencies.

We then describe our programming methodology and our novel middleware supporting distributed applications on constrained devices. Our development approach is based on distributed symbolic processing through executable code exchange among nodes, and permits toextend the capabilities of nodes even after the deployment.

V. HARDWARE DESCRIPTION

1. NodeMCU:

A Node MCU is a combination of a node and microcontroller unit. It is an open platform for edit/modify. It is permanentsoftware with read only memory and isuseful in prototyping.

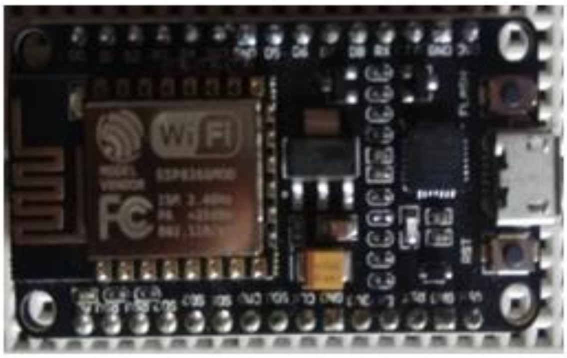


Fig 2. Node MCU

The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. Node MCU is similar to an Arduino with ESP8266 as its major component. It has Programmable pins and built in Wi-Fi. It can get power through micro-usb port. Its cost is low and can be programmed through multiple programming environments.

2. Jumper Wires:

Jumper pins (points to be connected by the jumper) are arranged in groups called jumper blocks, each group having at least one pair of contact points. An appropriately sized conductive sleeve called a jumper, or more technically, a shunt jumper, is slipped over the pins to complete the circuit.Jumpers must be electrically conducting; they are usually encased in a non-conductive block of plastic for convenience. This also avoids the risk that an unshielded jumper will accidentally short out something critical (particularly if it is dropped on a live circuit).

Fig 3. Jumper wire.

3. Gas Sensor:

MQ 135 sensor is 4-pin multi-use sensor. It can be used to sense gas like benzene, alcohol, smoke. MQ135 sensor has an electrochemical sensor inside it, and this sensor is sensitive to a range of gasses are used at room temperature. You can alsoread about the other IoT sensors if you want to measure a particular gas.

Fig 4. MQ135 Gas Sensor.

VI. SOFTWARE DESCRIPTION

1. Firebase Services:

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in Realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data. The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync between your users in Realtime.The Realtime Database is really just one big JSON object that the developers can manage in Realtime.

2. Arduino IDE:

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to- serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to- serial converter.

VII. SYSTEM ARCHITECTURE

1. Cloud Function:

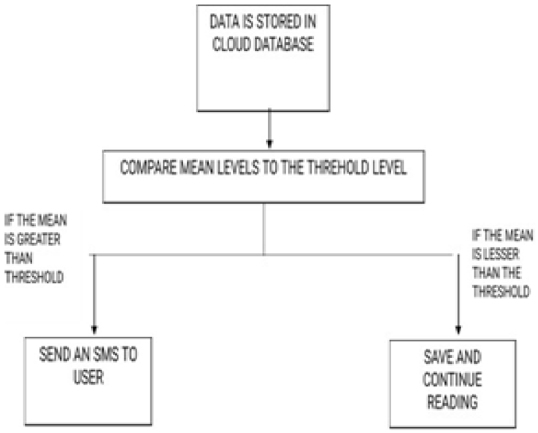

DATA IS STORED IN CLOUD DATABASE
COMPARE MEAN LEVELS TO THE THREHOLD LEVEL
IF THE MEAN IS GREATER THAN THRESHOLD
IF THE MEAN IS LESSER THAN THE THRESHOLD
SEND AN SMS TO USER
SAVE AND CONTINUE READING

Fig 5. Cloud Working.

1.1 Once the readings are stored in the database, these readings are fetched to the lambda function through the server.

1.2 In the lambda function we have a defined threshold value which is the normal range of any gas in air. The readings are added up to form a mean value which on comparison with the threshold:

- If greater than the threshold the Twilio functions are triggered and a SMS sent to the user
- If normal the device continues to read until the user decides to end the working

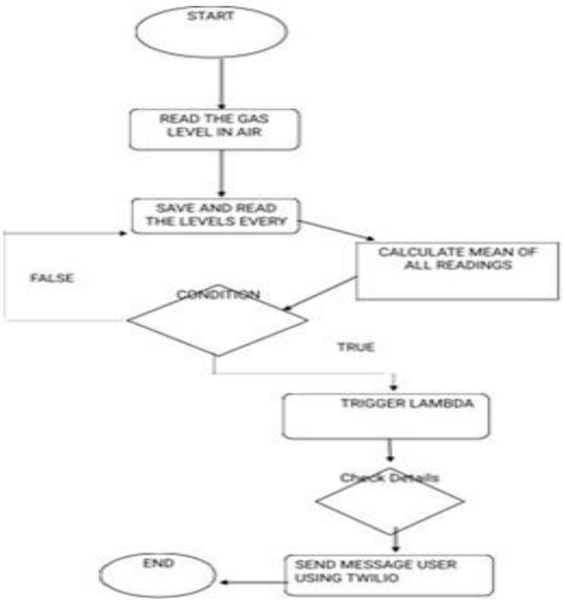

START
READ THE GAS LEVEL IN AIR
SAVE AND READ THE LEVELS EVERY
CALCULATE MEAN OF ALL READINGS
FALSE
CONDITION
TRUE
TRIGGER LAMBDA
Check Details
END
SEND MESSAGE USER USING TWILIO

Fig 6. Activity diagram.

2. Sequence Diagram:
These are the diagram which shows how objects operate with the one another object. Sequence construct with the messages and sequence chart visualization.

In diagram format the sequence diagrams are arranged. And it also describes the items and sequence of messages interchange between the different objects needed to perform certain task or situation.

The diagrams involved with different methods are used in the system or project. It shows the flow of the transactions done in the system. Here we use the two sequence diagram in this project one for the cloud function and user case.

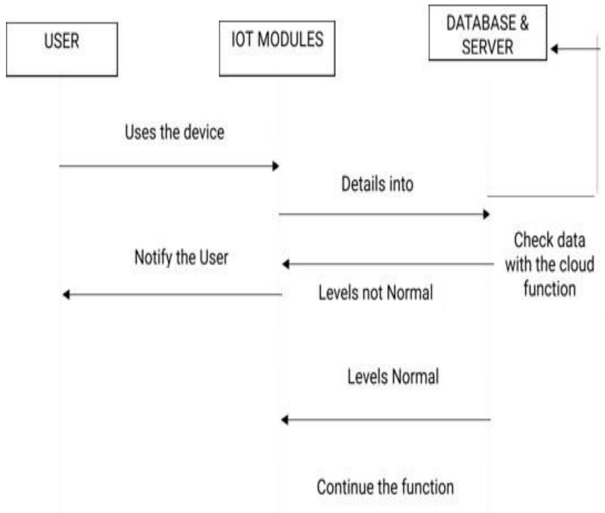


Fig 7. Sequence Diagram of user.

VIII. IMPLEMENTATION

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test is one that answers a yet undiscovered error. Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved.

The candidate system is subject to variety of tests-on-line response, volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to form, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mech”, that is the internal operation of the product performs according to the specifications and all internal components have been adequately exercised.

Types of testing:

- System testing
- Unit testing
- Integration testing
- Black box testing
- White box testing
- User acceptance testing
- Output testing

1. System Testing:
Here the entire software system is tested. The reference document for this process is the requirements document, and the goal OS to see a software meets its requirements. This project is tested in Linux OS and works well in this OS environment.

2. Unit Testing:
Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as module testing. The modules of the system are tested separately. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

3. Integration Testing:
Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance.
There are two type of integration testing. They are:

- Top-don integration testing
- Bottom-up

4. White Box Testing:
This is the unit testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors. We tested step wise every piece of code, taking care that every statement in the code is executed at least once, the white box testing is also called GLASS BOX Testing.

5. Black Box Testing:
This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details as statement level. Here the module will be treated as a black box that will take some input and generate output. Output for a given set of input combinations are forwarded to other module. We have performed black box testing by taking different

combinations of inputs such that the input passed will be transferred to different modules and is used correctly.

6. User Acceptance Testing:

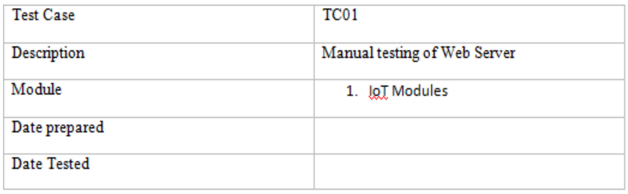
User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

7. Output Testing:

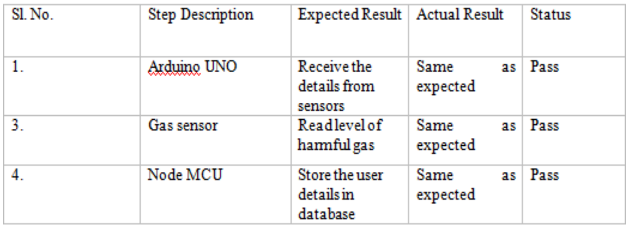
After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

8. Test Cases:

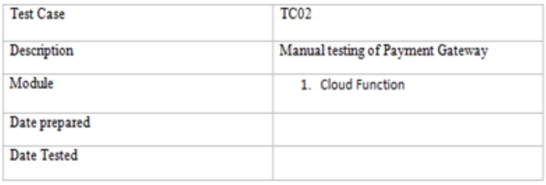
The system is configured to read the values of different sensors kept at different locations. Features to be tested:
Verifying the IoT Modules



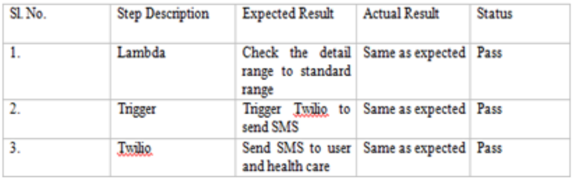
Test Case	TC01
Description	Manual testing of Web Server
Module	1. IoT Modules
Date prepared	
Date Tested	



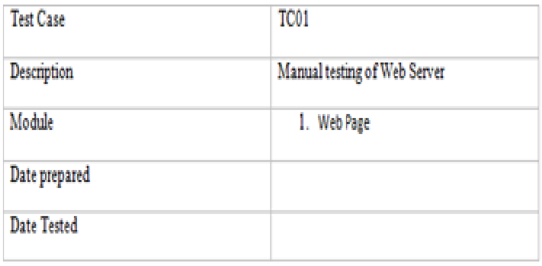
Sl. No.	Step Description	Expected Result	Actual Result	Status
1.	Arduino UNO	Receive the details from sensors	Same as expected	Pass
3.	Gas sensor	Read level of harmful gas	Same as expected	Pass
4.	Node MCU	Store the user details in database	Same as expected	Pass



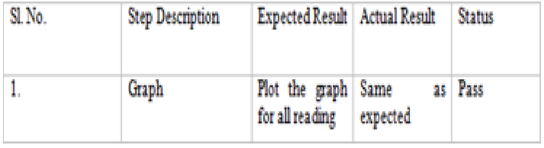
Test Case	TC02
Description	Manual testing of Payment Gateway
Module	1. Cloud Function
Date prepared	
Date Tested	



Sl. No.	Step Description	Expected Result	Actual Result	Status
1.	Lambda	Check the detail range to standard range	Same as expected	Pass
2.	Trigger	Trigger Twilio to send SMS	Same as expected	Pass
3.	Twilio	Send SMS to user and health care	Same as expected	Pass



Test Case	TC01
Description	Manual testing of Web Server
Module	1. Web Page
Date prepared	
Date Tested	



Sl. No.	Step Description	Expected Result	Actual Result	Status
1.	Graph	Plot the graph for all reading	Same as expected	Pass

IX. RESULT

The IoT module implemented reads the gases level from the air. This details is now stored into the database server. After the data is stored Lambda is implemented and Twilio is triggered to send a notification to the user. All these functionality is implemented and working in the same order.

X. CONCLUSION

Pollution in earlier days was negligible. Currently, however, pollution is increasing day-by-day because of various reasons such as industrial growth, development of automobile industries, and chemical industries. Therefore, to reduce the level of pollution from such sources and to protect humans and the environment from harmful gasses, this air pollution kit was developed that helps a person to detect, monitor, and test air pollution in a given area. The kit has been integrated with the SMS trigger that helps the user in predicting the pollution level of their entire route. Further, data logging can be used to predict AQI levels.

This proposed air pollution monitoring kit along with the integrated mobile application can be helpful to people suffering from respiratory diseases. The device has following features, indices of air quality for a specific city using real-time computation, air quality daily forecasts, timing outdoor activities for different recommendation of generation, air quality dips related to health risks, specific reports for air quality measures based on locations, air quality maps generation.

The proposed system faces with computational complexity particularly when we are dealing with big sensor data. One solution could be using fog computing, instead of cloud computing to reduce computation complexity and enhance the performance of the system. We can also implement zero tolerance fast big data real-time stream analytical tools to process such a complex system.

REFERENCES

[1] U. Varshney, Pervasive healthcare computing: EMR/EHR, wireless and health monitoring. Springer Science & Business Media, 2009.

[2] J. J. Caubel, T. E. Cados, and T. W. Kirchstetter, “A New Black Carbon Sensor for Dense Air Quality Monitoring Networks,” Sensors, vol. 18, no. 3, p. 738, 2018.

[3] L. Morawska, P. Thai, X. Liu, A. Asumadu-Sakyia, G. Ayoko, A. Bartonova, A. Bedini, F. Chai, B. Christensen, and M. Dunbabin, “Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: how far have they gone?,” Environ. Int., 2018.

[4] D. Santi, E. Magnani, M. Michelangeli, R. Grassi, B. Vecchi, G. Pedroni, L. Roli, M. C. De Santis, E. Baraldi, and M. Setti, “Seasonal variation of semen parameters correlates with environmental temperature and air pollution: A big data analysis over 6 years,” Environ. Pollut. vol. 235, pp. 806– 813, 2018.

[5] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, “Field calibration of a cluster of low-cost commercially available sensors for air quality monitoring. Part B: NO, CO and CO2,” Sensors Actuators B Chem., vol. 238, pp. 706–715, 2017.

[6] N. Castell, F. R. Dauge, P. Schneider, M. Vogt, U. Lerner, B. Fishbain, D. Broday, and A. Bartonova, “Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates?,” Environ. Int., vol. 99, pp. 293–302, 2017.

[7] S. Gaglio, G. Lo Re, G. Martorella, D. Peri, and S. D. Vassallo, “Development of an IoT environmental monitoring application with a novel middleware for resource constrained devices,” in Proceedings of the 2nd Conference on Mobile and Information Technologies in Medicine (MobileMed 2014), 2014.