Original Article

Air Pollution Monitoring System For Various Cities Using Cloud Computing And The Internet Of Things

Veerabhadraswamy K M,

Assistant Professor,
Department of Electronics and Communication Engineering,
Government Engineering College, Ramanagara, Karnataka-562159, India,
Veerabhadraswamy.13@gmail.com
DOI: 10.47750/pnr.2019.10.01.14

Abstract

Air pollution occurs when there are concentrations of hazardous substances, also referred to as pollutants, in the atmosphere, which have the potential to cause damage to ecosystems, human health, and the environment. These contaminants might exist in the form of gasses, particles, or biological agents. India's main cities have a significant number of companies, power plants, and factories leads to air pollution. However, without an awareness of the amount of damage caused by pollution, it may be impossible to implement such rules and regulations. As a result, the purpose of this project is to develop an Artificial Intelligence (AI) and Internet of Things (IoT) design with cloud storage to monitor the level of air pollution in various cities. For AI, three different models Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Artificial Neural Network (ANN). For input data Particulate Matter (PM) sensors, CO2 sensors, radiation sensors, and visible dust sensors were all used in this investigation. The sensors are linked into a circuit, which is distributed throughout India. AI model is developed using Tensorflow and based on the accuracy the best model is also integrated with the developed system of sensors and microcontrollers. To make this process easier for everyone to access and understand, a website is also developed. This website allows users to select a particular city or even add a new one. When the city is selected, the values of the parameters recorded from the sensors are displayed along with the level of air pollution. When one of the parameters exceeds the desired range, the website sends an alert notification to alert the officials about this. This process makes it easier for the officials to act accordingly and reduce the impact of pollution.

Keywords— Pollution, Internet Of Things, Cloud Computing, Sensors, Tensorflow, Web Development, Machine Learning.

I. INTRODUCTION

When dangerous compounds, often referred to as pollutants, are present in the air in amounts that could have a negative impact on ecosystems, the environment, and human health, this is referred to as air pollution [1]. Gases, particles, or biological agents can all be considered forms of these contaminants. The combustion of fossil fuels, construction operations, car exhaust, agricultural practices, and natural phenomena are all sources of air pollution [2, 3]. Air pollution has developed into a serious issue with detrimental effects in India's main cities. India's biggest cities are home to large concentrations of companies, power plants, and manufacturing facilities [4]. Building and infrastructure development projects produce dust and fumes from machinery, which increases the amount of PM in the air [5]. The use of emission standards, the promotion of cleaner technology, stronger vehicle laws, the use of alternative fuels, improved public transit, waste management programs, and public awareness campaigns are some of the actions taken to reduce air pollution in India's major cities. However, enforcing such rules and regulations might be impossible without knowing the level of damage the pollution has caused. So, this study aims to design an AI and IoT design with cloud storage to monitor the level of air pollution in different cities. The design and the detailed working of this circuit discussed elaborately in the upcoming chapters.

II. LITERATURE SURVEY

This literature review gives an in-depth summary of several research publications linked to enhanced air pollution monitoring systems. Here is a detailed examination of various papers. The paper [6] proposes an indoor air quality monitoring system with a real-time sensing unit capable of remotely monitoring carbon dioxide, humidity, and temperature. An IoT signal processor communicates with a low-power microcontroller in the system, which uses IoT technology. Its capabilities are expanded by the installation of a remote surveillance system based on Simple Mail Transfer Protocol and an immediate warning system focused on Short Messaging Service (SMS) through the SIM900 GSM module. The system responds quickly to distributing alarm messages and emails. It enables long-term data processing and logging, making it ideal for monitoring indoor air quality. The paper [7] describes an Ethernet-based remote control and monitoring system based on the LM3S8962 microcontroller. The system is notable for its stability and extremely low power usage. It facilitates the building of a handmade weather station with internet access by using the Lightweight IP (LwIP) protocol, which automatically obtains an IP address. Users can access the system's web server via mobile phone or PC web browsers, which shows data on parameters such as temperature, humidity, voltage, and PM2.5. Furthermore, the system allows sub-node peripherals to be controlled via control instructions that are uploaded and delivered to the sub-nodes. It enables wireless LAN to internet connectivity for remote monitoring and control via ZigBee. The local LCD panel displays critical parameters as well as IP addresses. This machine also acts as a client, uploading data to Yeelink, a typical data center networking platform. The research [8] proposes a novel strategy for developing air quality monitoring systems utilizing cutting-edge IoT methods. Real-time air quality data is collected using portable sensors and delivered via a low-power wide area network. In the IoT cloud, data is processed and analyzed. The finished air quality monitoring system, which includes both hardware and software components, has been successfully deployed in urban areas. The experimental results show the system's dependability in sensing air quality, which aids in the discovery of patterns and variations in air quality. This technology is ideal for monitoring air pollution in cities.

The research [9] describes a unique real-time air pollution monitoring and forecasting system that uses IoT technologies to preserve the environment. The system's hardware expenses are dramatically reduced by embracing IoT, making it more cost-effective and scalable. It can be used to construct a network of monitoring sensors in vast monitoring areas. Aside from standard air monitoring capabilities, the system is notable for its capacity to forecast the development trends of air pollution over a specific time period. This forecasting is accomplished by using neural network technology to analyze data provided from a front-end perception system. In practice, this capability enables targeted emergency steps to be performed in order to limit losses and alleviate the impact of air pollution occurrences. The paper [10] focuses on the design and implementation of an IoT-based Wi-Fi-based plug-and-sense smart device for dedicated air pollution monitoring. For accurate air pollution monitoring, the system employs a device-to-cloud architecture. Sensors within the device measure individual pollutant components and location coordinates, allowing the Air Quality Index (AQI) to be calculated. The AQI is calculated using a linear segmentation approach, the Greater Vancouver AQI dataset, and a Max operator aggregation method. LEDs are used in the system to indicate air quality levels and provide health impact information. Precautionary messages are also presented on the screen. All collected data is sent to ThingSpeak, an open-source IoT application programming interface. This information, together with timestamps, can be exported to a separate Excel file for further study. A ThingView Android app visualizes pollution levels by location in real time using line graphs. This low-cost and compact smart device improves public safety by delivering alerts, urging people to use anti-pollution masks, and recommending alternative transportation routes in high-pollution locations, while assuring high dependability and consistency.

A low-cost mobile device for measuring air quality indicators is described in the paper [11]. It employs a multilayer distributed framework, with an Arduino platform functioning as an IoT application. In this research, a physical device is furnished with software, electronics, sensor and wireless connection to monitor air pollution in real world. The electrical devices has three sensors that monitor CO, CO2, and powder density levels. This data collecting is facilitated by an API written in the C++ programming language. This approach was validated in several Ecuadorian cities, including Quito, Amaguaa, and Tena. The results of air pollutant concentrations acquired by this approach are consistent with the reference levels provided by international environmental organizations. This technology provides a cost-effective solution to air quality monitoring while ensuring that the data collected meets international requirements. The primary goal of this paper [12] is to present a vehicle pollution monitoring system that makes use of IoT technologies. This technology is intended to detect polluting automobiles on city streets and to measure various sorts of pollutants in the air. Furthermore, it notifies the state of air quality to environmental agencies as needed. The suggested system ensures the presence of wireless sensors specializing in vehicle-related pollution monitoring. Using IoT, these sensors allow simple real-time data access via the internet. The measured data is shared not just with environmental organizations, but also with car owners and traffic departments. This low-cost technology has been shown to be effective in reducing air pollution, particularly in cities. It addresses the essential issue of automobile pollution, assisting in the reduction of its negative influence on air quality and public health.

III. MATERIALS AND METHODS

The main objective of the study is to develop a system with AI, IoT and an assortment of sensors. Figure 1 explains the circuit design and the workflow in this process.

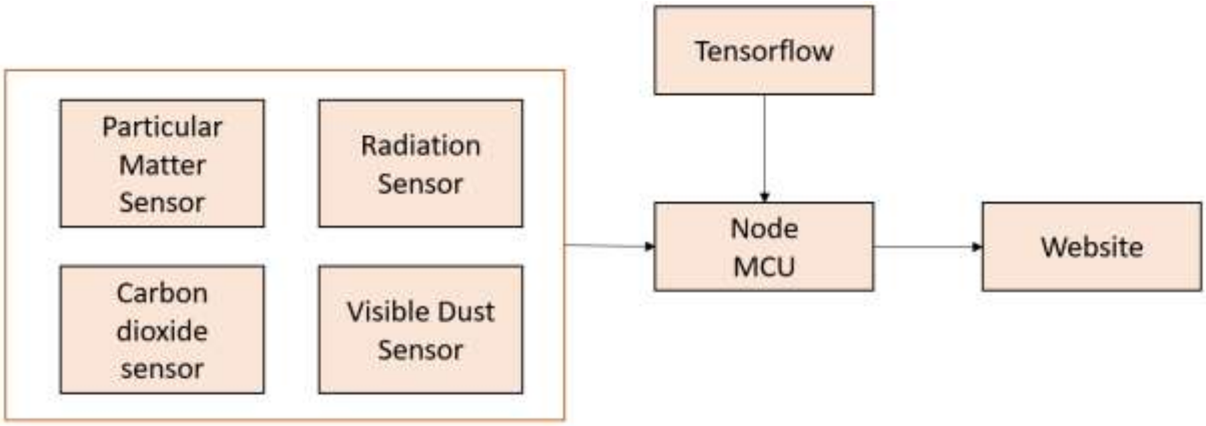

Figure 1. Workflow of the process

From Figure 1, it can be seen that the sensor combination is first chosen. This experiment made use of the particular matter sensor, CO2 sensor, radiation sensor, and visible dust sensor. The sensors are connected by wire to create a circuit that is set up all over different Indian towns. Tensorflow is incorporated with the developed system of sensors and microcontrollers in the construction of cloud storage. To make it easier for everyone to access and understand this process, a website has also been developed. The user of this website can select a certain city or add a new city. When the city is selected, the status of the level of air pollution is displayed together with the values of the metrics that are gleaned from the sensors.

IV. HARDWARE COMPONENTS

The sensors play a crucial role in this study as it is required for proper input to process and alert the officials about the presence of an unusual amount of air pollution of various forms like dust, and radiation. In this study, four different sensors are used. They are the particular matter sensor, CO2 sensor, radiation sensor, and visible dust sensor. All the sensors are controlled using a microcontroller. The NodeMCU module is used in this study. The working principle of all these components is explained clearly in this chapter.

A. Particular Matter (PM) Sensor

A PM sensor detects and measures the concentration of PM in the air. PM are tiny solid or liquid particles suspended in the air that can come from a range of sources, including industrial pollution, automobile exhaust, as well as dust storms [13]. Through the intake, the sensor draws outside air into the apparatus. The detecting element and the particulate materials inside the sensor interact. Gravimetric sensors collect particles on a filter, electrical sensors measure changes in electrical characteristics, and optical sensors measure scattered or absorbed light. A signal corresponding to the PM concentration is generated by the sensor's detection mechanism. For monitoring or additional analysis, the sensor output can be employed.

B. CO2 Sensor

Carbon dioxide gas concentration in the immediate surroundings is measured using a carbon dioxide (CO_2) sensor. CO_2 sensors are used in a variety of industries, including environmental research, HVAC systems, industrial operations, and indoor air quality monitoring. Non-dispersive infrared (NDIR) and electrochemical sensors are the two most used kinds of CO_2 sensors, and they can be made using a variety of methods. Due to their accuracy and stability, NDIR sensors are frequently employed for CO_2 detection. Chemical interaction between CO_2 and an electrolyte is the basis for how electrochemical CO_2 sensors work [14]. The carbon dioxide-containing ambient air is exposed to the CO_2 sensor. To get to the sensing mechanism, gas diffuses into the sensor. Digital data, analogue voltage, or a display showing the CO_2 concentration directly are just a few of the output formats that a sensor can produce.

C. Radiation sensor

Ionizing radiation is detected and measured by a radiation sensor, which is also known as a radiation detector or dosimeter [15]. Radiation protection, nuclear power plants, healthcare facilities, and environmental monitoring are only a few examples of its widespread use. Ionising radiation enters a sensor and reacts with the detection medium, which may be a gas, scintillator, or semiconductor. Ion pairs are created or excited atoms or molecules as a result of radiation's transfer of energy to the detecting medium. Depending on the type of sensor, these processes discharge charged particles, light photons, or electron-hole pairs. Depending on the sensor design, the released charged particles, photons, or electron-hole pairs provide an electrical or optical signal. In order to improve the signal-to-noise ratio and recover valuable information, signals are often amplified and processed by electronics.

D. Visible dust sensor

A device used to identify and gauge the amount of dust in the air is called a dust particle sensor, sometimes known as a dust sensor. These sensors are frequently employed in industrial applications, environmental monitoring, and indoor air quality monitoring. Into its sensing chamber or sample region, the sensor sucks in ambient air. Whether it be through light scattering, gravimetric deposition on a filter, or variations in capacitance, the airborne dust particles interact with the sensor's detection mechanism [16]. The electronics of the sensor analyse the signals produced by the sensing mechanism to calculate the size and density of the dust particles. The sensor outputs data that can take on a number of different shapes, including digital data, analogue voltage, or a display of the dust particle concentration.

E. NodeMCU controller

NodeMCU is a development board with open-source firmware that depends on the ESP8266 chip. The ESP8266 Wi-Fi chip, which is inexpensive, makes it simple to include Wi-Fi connectivity in a variety of applications, including those pertaining to the IoT. For the quick prototyping and development of IoT applications, NodeMCU offers a software platform and a hardware platform. Because NodeMCU has built-in Wi-Fi, it can connect to Wi-Fi networks and use the internet to communicate with other devices and cloud services. The Lua scripting language is supported by the NodeMCU firmware, making it simple to write and run code on a device. AWS IoT, Google Cloud IoT Core, and other cloud platforms and services are just a few of the ones that NodeMCU can interact with. This makes it possible for the NodeMCU board and cloud-based apps to communicate seamlessly, enabling remote monitoring, data analysis and control.

V. SOFTWARE COMPONENTS

The data collected from the sensors are analyzed using the microcontroller module. But for that process to happen, the data had to be stored somewhere. In this study, the Tensorflow is used for developing an AI model. Google offers a service called TensorFlow Cloud that makes it easier to train and use TensorFlow models in the cloud. Utilizing the scalability and resources of cloud infrastructure provides a streamlined approach for managing machine learning trials. It facilitates team cooperation, improves scalability, and makes experiment management simpler. TensorFlow Cloud takes advantage of cloud infrastructure and services to free users from worrying about infrastructure maintenance so they can concentrate on creating and enhancing their machine learning models.

A. ANN

ANN [17] can be seen as parallel and distributed processing systems comprised of numerous simple, highly interconnected processors. Among the various architectural paradigms in ANN today, the Multilayer Perceptron (MLP) is the most widely used. All neurons inside a layer are associated with all neurons in subsequent layers via unidirectional connections, which is a common feature of neural networks of this class. These connections only send data in the forward direction. Each connection is associated with adjustable weights, which are modified based on a defined learning rule.

The training of feedforward neural networks typically employs the Backpropagation Algorithm (BPA) [18]. The network creates a nonlinear mapping among input and output variables using this approach. As a result, with provided input/output pairs, the BPA modifies the weights of the network to represent the nonlinear relationship. After training, a network with fixed weights can generate output for a given input. The conventional BPA for network training attempts to reduce an energy function representing immediate error. In essence, the purpose is to reduce the following function:

$$E(m) = \frac{1}{2}\sum_{q=1}^{n}[d_q - y_q]^2 \quad [1]$$

Here, d_q signifies the required output for the q^{th} input, and y_q represents the actual neural network output. The adjustment of each weight follows the rule:

$$\Delta w_{ij} = -k\frac{dE}{dw_{ij}} \quad [2]$$

In this equation, E is the error function, k is a proportionality constant, and w_{ij} represents the weights between neurons j and i. The weighting procedure is repeated until the variation between the node and the actual output is within a permissible range.

B. KNN

KNN represents a straightforward algorithm that stores the available cases and makes classifications for new instances based on similarity measures, often involving distance functions [19]. KNN has a historical presence in statistical estimation and pattern recognition dating back to the early 1970s, recognized for its non-parametric nature.

The functioning of the KNN algorithm is as follows: when assigning a class to a particular case, it relies on the majority vote from its neighboring cases. The case is then categorized into the class that is most prevalent among its K nearest neighbors, as determined by a distance function. If K equals 1, the case straightforwardly belongs to the class of its closest neighbor. There are several distance metrics applicable to continuous variables, such as Euclidean, Manhattan, and Minkowski distances. In our context, we have employed the Euclidean distance formula, which quantifies the distance between data points in multi-dimensional space.

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \quad [3]$$

Selecting the optimal value for K is a task best tackled by initially scrutinizing the dataset. Generally, opting for a larger K value enhances precision by mitigating the impact of random noise, but it is essential to acknowledge that there is no certainty that the largest K will consistently yield the best results. Alternatively, cross-validation presents a retrospective method for pinpointing an appropriate K value, often within the range of 3 to 10. This process balances the risks of overfitting and the quest for accurate classification, ensuring that KNN performs optimally.

C. SVM

In 1995, SVM became a frequently used categorization algorithm. In categorization, the conventional process entails separating data into training and testing samples, with each instance in the training sample including a "target" and an array of "attributes." The fundamental goal of SVM is to build a model using training sample that will forecast the target values of the test data based on the features [20]. For a given training set of instance-label pairs (x_i, y_i) $i = 1, \ldots, l$ where $x_i \in R^n$ and $y \in \{1\text{-}1\}^l$, SVM necessitates the solution of the following optimization problem:

$$\min \quad \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \quad [4]$$

$$\text{Subject to} \quad y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \quad [5]$$

The function is used to map the training vectors x_i onto a higher dimensional space. In this higher-dimensional space, SVM aims to locate a linear separating hyperplane with the greatest margin. $C > 0$ acts as the penalty variable for the error term. Furthermore, the kernel function is represented as $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$, and there are four basic kernels: polynomial, linear, radial basis function (RBF), and sigmoid. We used the RBF kernel for our application:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0 \quad [6]$$

VI. Result And Discussion

The study aims to design a circuit consisting of sensors, microcontrollers, and AI in various cities to constantly monitor the pollution level and alert the officials in case of increases in pollution level. However, it cannot be possible for someone to constantly monitor the levels to check whether the values stay within the range or not. Thus, a website that displays all the output values from the sensors of all the listed locations is designed. The figure 2 summarizes the findings of the air pollution monitoring system that used ML methods such as SVM, KNN, and ANN. The models' accuracy is outstanding, with ANN achieving the highest accuracy at 98.33%, closely followed by SVM at 97.33%, and KNN at 94.67%. These findings highlight the accuracy with which these methods classify air pollution data. Specificity and sensitivity, which assess the model's ability to properly distinguish non-polluted and polluted cases, also show that SVM and ANN perform well. SVM has a high specificity of 98.64% and an acceptable sensitivity of 96.08%. ANN produces excellent results, with a high specificity (99.32%) and sensitivity (97.37%). SVM and ANN both outperform in terms of positive predictive value (PPV), with SVM scoring 98.66% and ANN scoring 99.33%. Finally, negative predictive value (NPV) demonstrates SVM and ANN's great performance in reliably predicting negative instances, with SVM achieving 96.03% NPV and ANN achieving 97.35%. KNN, while trailing in several metrics, nonetheless produces respectable results. ANN excel at identifying air pollution data, with high accuracy and predictive values, making them ideal tools for air quality assessment and monitoring.

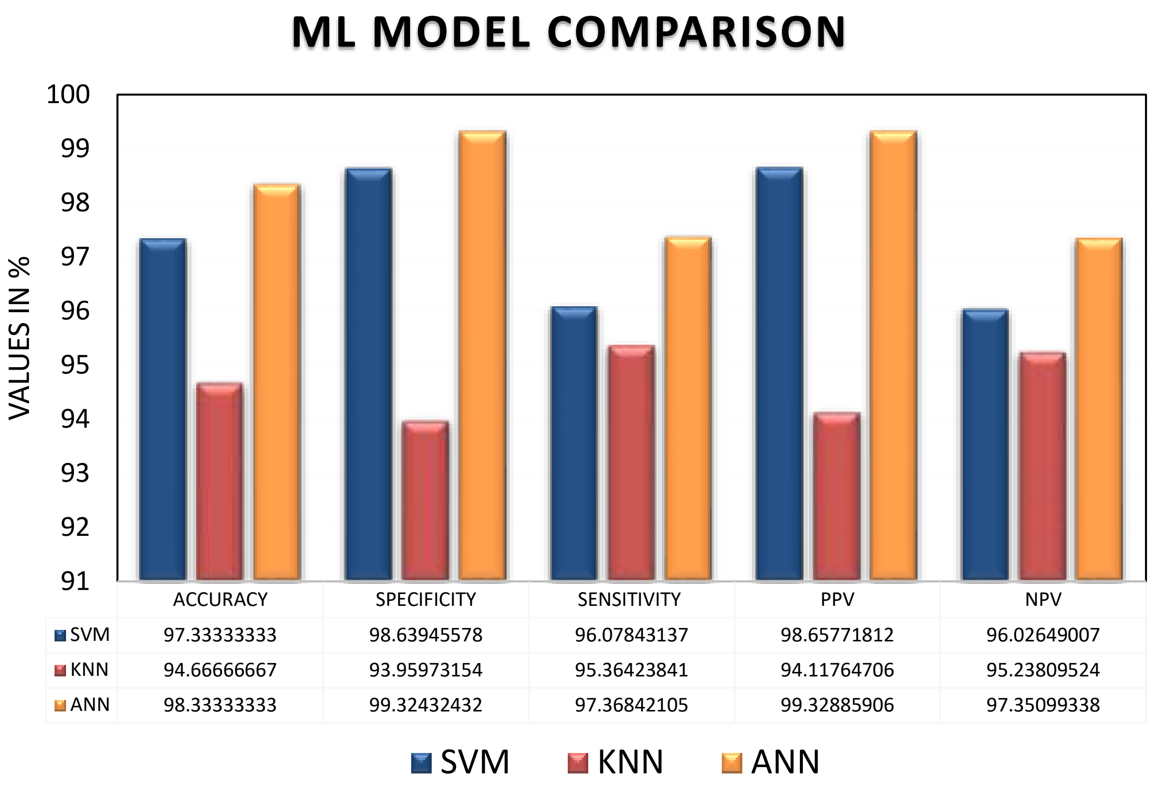

Figure 2. ML model performance on air pollution monitoring system

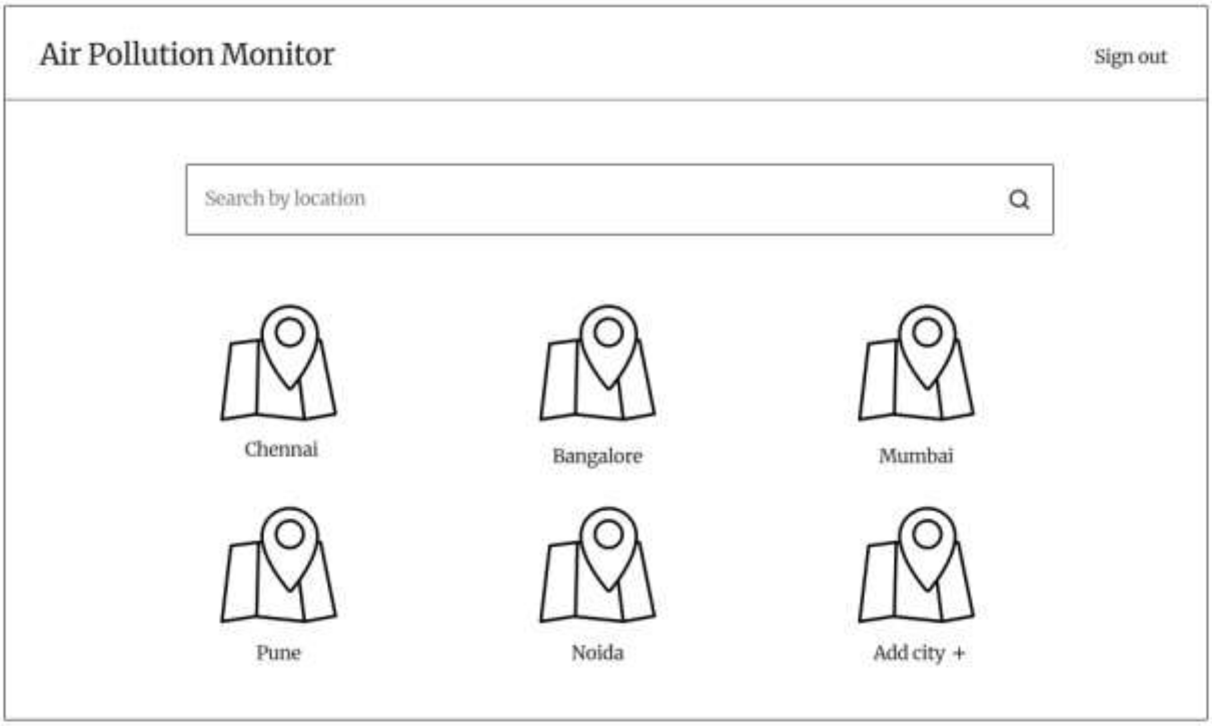

Figure 3. Home page of the website

The homepage of the website is shown in Figure 3. As shown in Figure, the header of the website contains the name of the website and a sign-out button. The sign-out button indicates that the user is currently signed in and only then it displays the list of cities. The main part of the website consists of a search bar where the user can search for a particular location. Below the search bar, a set of icons each indicating a city is displayed. It also consists of an Add City button which allows the user to enter a new city with a new set of calibrated sensors which are already present in that particular city. So, when the user selects one of the existing cities, the website displays the details about the city. The detailed report of a particular city is shown in Figure 4.

In Figure 4, the city which is selected is displayed. Four different values – the particular matter, CO2 levels, radiation level and visible dust level are shown also displayed here. The values are collected from the sensors which are placed in that particular city itself. Below the values, a status which indicates that all the parameters are within a safe limit is also shown for easier reference. When one of the values exceeds the limit, an alert message is sent by the website and it is shown in Figure 5.

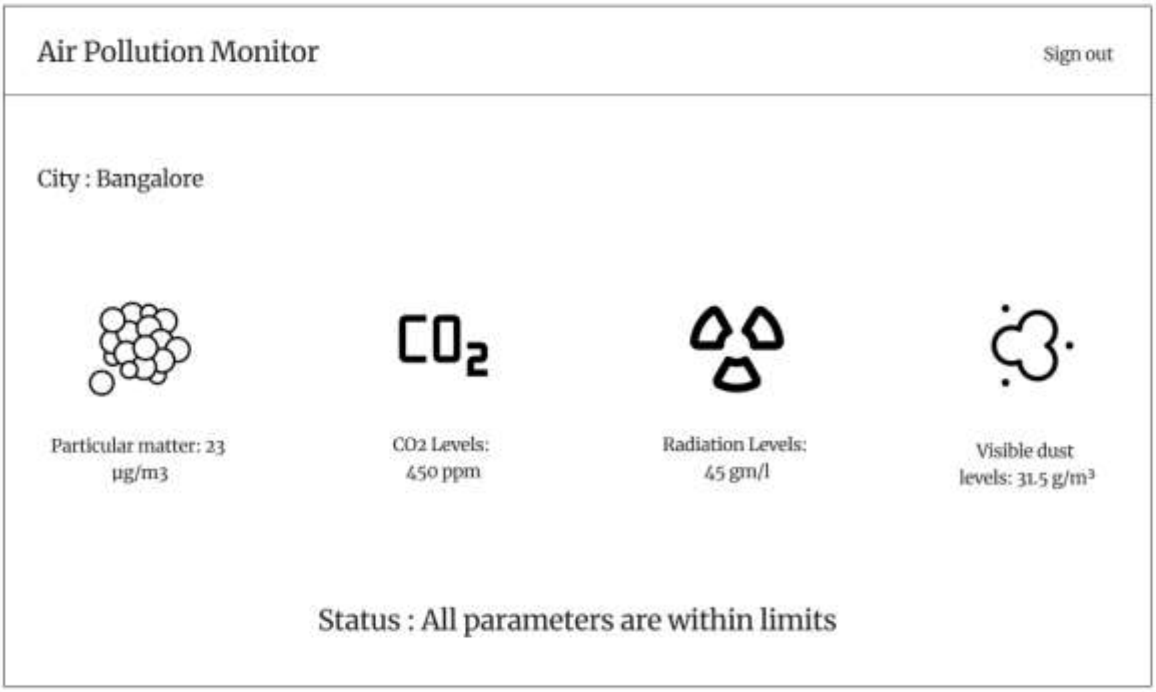

Figure 4. Detailed report of a city

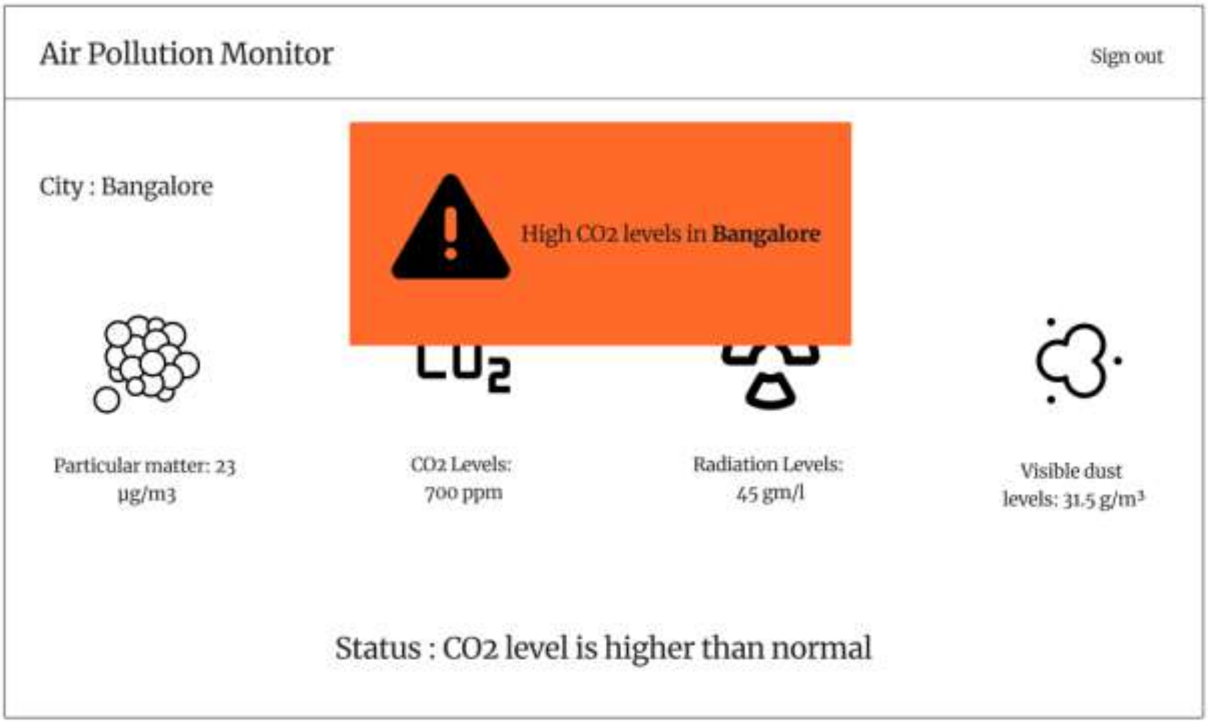

Figure 5. Alert message from the website

In the figure above it can be seen that the CO2 levels are higher than the safe limit. Thus, the status of the website is changed to an alert message along with a pop-up alert message. Here, as the entire website is monochromatic and the alert message is in red, it immediately attracts the attention of the officials so they can take any measures that can reduce the CO2 levels in that particular region.

VII. CONCLUSION

Air pollution is caused by the burning of fossil fuels, building activities, car exhaust, agricultural activities, and natural occurrences. In India's major cities, air pollution has become a big problem with negative impacts. Some of the steps made to lower air pollution in India's largest cities include the adoption of emission regulations, the promotion of cleaner technologies, stricter car laws, the use of alternative fuels, improved public transit, waste management initiatives, and public awareness programs. Yet, without understanding the extent of the harm that pollution has done, it may be impossible to enforce such rules and regulations. A circuit was designed in this study to assist officials by notifying the fact that air pollution has gone bad in a particular region. Firstly, a sensor combination is picked. The specific matter sensor, CO2 sensor, radiation sensor, and visible dust sensor were all used in this investigation. The sensors are wired together to form a circuit, which is installed throughout several Indian cities. Tensorflow is used in the development of ML model, and it is combined with the created system of sensors and microcontrollers. A website is also created to make it simpler for everyone to access and comprehend this procedure. The user of this website has the option to add a new city or choose a specific city. The status of the degree of air pollution is shown together with the values of the metrics that are recorded from the sensors when the city is selected. In the future, the website can be updated into a more secure one by adding administration rights to the officials and user rights to the general public so everyone can be aware of the state of pollution in any city they wanted.

REFERENCE

[1]. Mannucci, Pier Mannuccio, Sergio Harari, Ida Martinelli, and Massimo Franchini. "Effects on health of air pollution: a narrative review." Internal and emergency medicine 10 (2015): 657-662.
[2]. Sivaramanan, Sivakumaran. "Air Pollution sources, pollutants and mitigation measures." Research Gate, DOI 10, no. 2.1 (2015): 5106-8485.
[3]. Sonwani, Saurabh, and Pallavi Saxena. "Identifying the sources of primary air pollutants and their impact on environmental health: a review." IJETR 6, no. 2 (2016): 111-130.
[4]. Chandra, Mahesh. "Environmental concerns in India: Problems and solutions." J. Int'l Bus. & L. 15 (2015): 1.
[5]. Wu, Zezhou, Xiaoling Zhang, and Min Wu. "Mitigating construction dust pollution: State of the art and the way forward." Journal of cleaner production 112 (2016): 1658-1666.
[6]. Girish, Siva V., R. Prakash, and A. Balaji Ganesh. "Real-time remote monitoring of indoor air quality using Internet of Things (IoT) and GSM connectivity." In Artificial Intelligence and Evolutionary Computations in Engineering Systems: Proceedings of ICAIECES 2015, pp. 527-533. Springer India, 2016.
[7]. Wang, Dongyun, Chenglong Jiang, and Yongping Dan. "Design of air quality monitoring system based on Internet of Things." In 2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA), pp. 418-423. IEEE, 2016.
[8]. Zheng, Kan, Shaohang Zhao, Zhe Yang, Xiong Xiong, and Wei Xiang. "Design and implementation of LPWA-based air quality monitoring system." IEEE Access 4 (2016): 3238-3245.
[9]. Xiaojun, Chen, Liu Xianpeng, and Xu Peng. "IOT-based air pollution monitoring and forecasting system." In 2015 international conference on computer and computational sciences (ICCCS), pp. 257-260. IEEE, 2015.
[10]. Gokul, V., and Sitaram Tadepalli. "Implementation of a WiFi based plug and sense device for dedicated air pollution monitoring using IoT." In 2016 Online International Conference on Green Engineering and Technologies (IC-GET), pp. 1-7. IEEE, 2016.
[11]. W. Fuertes et al., "Distributed System as Internet of Things for a New Low-Cost, Air Pollution Wireless Monitoring on Real Time," 2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Chengdu, China, 2015, pp. 58-67, doi: 10.1109/DS-RT.2015.28.
[12]. Rushikesh, Ramagiri, and Chandra Mohan Reddy Sivappagari. "Development of IoT based vehicular pollution monitoring system." In 2015 international conference on green computing and internet of things (ICGCIoT), pp. 779-783. IEEE, 2015.
[13]. Jovašević-Stojanović, Milena, Alena Bartonova, Dušan Topalović, Ivan Lazović, Boris Pokrić, and Zoran Ristovski. "On the use of small and cheaper sensors and devices for indicative citizen-based monitoring of respirable particulate matter." Environmental Pollution 206 (2015): 696-704.
[14]. Behera, Kamalakanta, Shubha Pandey, Anu Kadyan, and Siddharth Pandey. "Ionic liquid-based optical and electrochemical carbon dioxide sensors." Sensors 15, no. 12 (2015): 30487-30503.
[15]. Silva, Marcia Dutra R. "Ionizing radiation detectors." Evolution of Ionizing Radiation Research (2015): 189-209.
[16]. Shaik, Mubeena, Naseema Shaik, and Wali Ullah Shaik. "The wireless sensor networks: smart dust." International Research Journal of Engineering and Technology 3, no. 6 (2016): 910-913.
[17]. Kukreja, Harsh, N. Bharath, C. S. Siddesh, and S. Kuldeep. "An introduction to artificial neural network." Int J Adv Res Innov Ideas Educ 1 (2016): 27-30.
[18]. Cilimkovic, Mirza. "Neural networks and back propagation algorithm." Institute of Technology Blanchardstown, Blanchardstown Road North Dublin 15, no. 1 (2015).
[19]. Zhang, Zhongheng. "Introduction to machine learning: k-nearest neighbors." Annals of translational medicine 4, no. 11 (2016).
[20]. Awad, Mariette, Rahul Khanna, Mariette Awad, and Rahul Khanna. "Support vector machines for classification." Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers (2015): 39-66.