IOT Based Real - Time Air Quality Monitoring System

Mr. S. Keerthipriyan[1], S. Mahalakshmi[2], M. Priyanka[3], M. Muthumanikandan[4]
[1,2,3,4]*AAA College Og Engineering and Technology*

Abstract—**The increasing concerns over air pollution and its impact on human health have led to the development of innovative air quality monitoring systems. This paper presents an IoT-enabled air quality monitoring system using an MQ series gas sensor for detecting various air pollutants, paired with a temperature sensor for environmental monitoring. The system is controlled by an Arduino microcontroller to ensure efficient data acquisition and processing. Sensor readings are analyzed to determine the severity of air quality, with a Multi-Layer Perceptron (MLP) algorithm used for classification. The MLP model categorizes air quality into different levels of severity, such as 'Good', 'Moderate', and 'Hazardous', providing a valuable tool for early detection of air pollution hazards. This proposed system offers a cost-effective, real-time solution for air quality monitoring, contributing to better public health management and environmental safety.**

Index Terms—**Air Quality Monitoring, Internet of Things (IoT), MQ Series Gas Sensor, Arduino Microcontroller, Temperature Sensor, Air Pollution Detection, Multi-Layer Perceptron (MLP), Machine Learning Classification, Real-Time Monitoring, Environmental Monitoring.**

1. INTRODUCTION

Air pollution has emerged as one of the most significant environmental challenges of the modern era, with detrimental effects on human health, climate change, and ecosystems. As urbanization continues to accelerate and industrial activities expand, the levels of harmful pollutants such as particulate matter (PM), nitrogen dioxide (NO_2), carbon monoxide (CO), and volatile organic compounds (VOCs) are increasing, posing serious risks to public health. Exposure to poor air quality has been linked to respiratory diseases, cardiovascular problems, and a range of chronic conditions, particularly in densely populated areas. The growing concern over air pollution's impact on both human health and the environment has spurred the need for more effective, accessible, and real-time air quality monitoring systems.

Traditional methods of air quality measurement often involve stationary monitoring stations that are expensive and require significant maintenance. While these systems provide valuable data, they are typically limited by their geographic scope and lack of real-time feedback. Additionally, they may not offer the granularity needed to capture the variability of air quality across different urban areas. In response to these limitations, there has been a surge in the development of Internet of Things (IoT)-based solutions, which integrate low-cost, portable sensors with real-time data transmission capabilities, making it possible to monitor air quality more effectively and efficiently. The potential for IoT-based air quality monitoring systems lies in their ability to provide continuous, real-time data that can be accessed remotely, enabling quick responses to hazardous air quality conditions. These systems leverage a network of interconnected sensors that collect various environmental parameters, including air pollutants, temperature, and humidity. The integration of machine learning models with these sensors further enhances the system's ability to classify air quality and predict pollution trends, offering critical insights for early intervention and policy decision-making.

With advancements in sensor technology and machine learning, these IoT-based systems are becoming more accurate, reliable, and cost-effective. By providing real-time, localized data on air quality, they empower communities to take proactive measures to mitigate the risks associated with pollution, improving both public health and environmental safety.

II. LITERATURE REVIEW

Jonathan M. Samet, M.D. et al [1] demonstrated in developed and developing countries, indoor air pollution is gaining increasing prominence as a public health problem. Exposures to indoor pollutants can impair physiological functioning, although not to a degree necessarily associated with disability or disease. Ivan Gee et al [2] study on the theme of monitoring indoor air pollution. Indoor pollution is not a new problem and Manchester, as one of the foremost cities in the industrial revolution, has in its past experienced many indoor pollution problems. Images of children crawling under cotton spinning machines thick with cotton dust graphically illustrate the many industrial problems of the time

Lidia Morawska et al [3] describes the purpose of exposure assessment as that the study design and the choice of the parameter measured should depend on the expected relation between exposure and health effects considered. W. Baldauf et al [4] proposed a model establishing ambient air quality monitoring sites based on potential health risks to the exposed population was developed. Steven M. Bortnicket al [5] describes the Environmental Protection Agency's data quality objectives process was employed to develop a measure of adequacy for deciding whether a statistical linear regression model that relates Federal Reference Method and continuous PM2.5measurements is sufficient for use in AQI reporting. Next, MSA-specific models were developed for NC and IA/IL that relate Federal Reference Method PM2.5 measurements with Continuous Monitoring measurements.

K. Papakonstantinou et al [6] presented simulation, calculation of CO percentage in different areas. In this study, dispersion of CO levels of the indoor environment in 2 different conditions i. e with mechanical ventilation of 2 m/s and nonmechanical ventilation. In this model CO levels conjunction with Phonics, CFD code that can give information of CO percentage, velocity fields in 3-dimensional configuration. K. Max Zhang et al [7] suggested about aerosol concentrations are correlated with health effects on people. Therefore, it is important to understand how particles transport and transform near roadways especially regarding the public health of those traveling on and living near freeways. In this work, we shall investigate the evolution of aerosol number distributions near roadways and elucidate the aerosol dynamics near mobile sources. A. J. Lawrence et al [8] study the association between the outdoor and indoor percentage of pollutants are observed in different areas. Statistical analysis was used to correlate indoor concentration levels without door concentrations of CO, NO, and NO2

III. PROPOSED SYSTEM

The proposed air quality monitoring system leverages IoT technology to create a smart, real-time solution for detecting and classifying air pollutants. The system is built around an MQ series gas sensor, capable of detecting a range of gases such as carbon monoxide (CO), nitrogen dioxide (NO_2), and methane (CH_4), which are critical indicators of air pollution. Additionally, a temperature sensor is integrated to account for environmental conditions, ensuring accurate readings across varying climates. The core of the system is an Arduino microcontroller, which handles the acquisition of data from the sensors and transmits it for processing. The collected sensor data is then processed through a Multi-Layer Perceptron (MLP) algorithm, which classifies the air quality into severity levels 'Good', 'Moderate', and 'Hazardous'. This classification helps in providing real-time alerts, allowing users to take preventive measures based on the air quality. The system is designed to be scalable and adaptable, making it a suitable solution for urban areas, industrial zones, or even for individual homes looking to monitor air quality effectively.

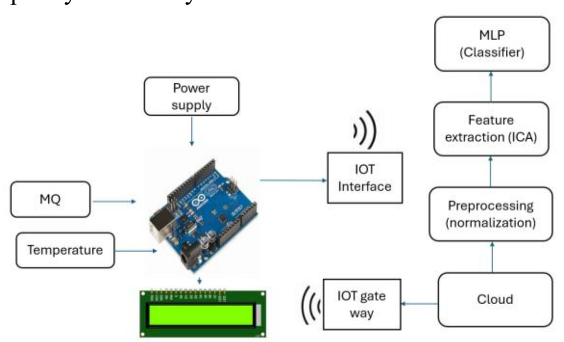

Figure Proposed system

1. Sensor Module:

The sensor module of the proposed system includes an MQ series gas sensor and a temperature sensor. The MQ gas sensor is designed to detect a variety of gases such as carbon monoxide (CO), nitrogen dioxide (NO_2), and methane (CH_4), which are key indicators of air quality. This sensor provides real-time data on the concentration of gases in the environment, enabling continuous monitoring. The temperature sensor is integrated to account for environmental fluctuations that might affect the gas sensor's performance. Together, these sensors provide comprehensive data about the air quality, which is essential for accurate monitoring and decision-making.

2. Data Acquisition Module:
The data acquisition module is based on the Arduino microcontroller, which serves as the central controller for the system. The Arduino collects data from the gas and temperature sensors, processes the signals, and sends them to a connected server or cloud platform for further analysis. The microcontroller is programmed to ensure seamless communication between the sensors and the system, offering a low-cost and reliable platform for data collection. The module can be expanded to integrate additional sensors, such as particulate matter or humidity sensors, as needed.

3. Data Processing Module:
The data processing module utilizes the collected sensor data to assess the air quality. The sensor data is input into a machine learning model, specifically a Multi-Layer Perceptron (MLP) algorithm. This algorithm classifies the air quality based on predefined categories, such as 'Good', 'Moderate', and 'Hazardous'. The MLP model is trained using historical air quality data to recognize patterns in the sensor readings and determine the severity of the air quality. The processed data is then forwarded to the output module for visualization and user interaction.

4. Classification Module:
The classification module, built using the MLP algorithm, plays a pivotal role in determining the air quality severity. After processing the raw sensor data, the MLP model classifies the air quality into different levels based on the gas concentration and environmental conditions. For example, if the gas concentrations exceed a certain threshold, the system will classify the air quality as 'Hazardous'. If the levels are moderate, it will categorize the air quality as 'Moderate'. This classification is crucial for providing early warnings and enabling timely actions to protect public health.

5. Alert and Notification Module:
The alert and notification module is designed to provide real-time notifications when the air quality reaches dangerous levels. The system is programmed to send alerts to users through multiple channels, such as SMS, email, or app notifications. This ensures that individuals, especially those with respiratory conditions, are informed of air quality changes and can take preventive measures. The system can also integrate with home automation setups, triggering actions such as turning on air purifiers or ventilating the space when hazardous air quality is detected.

6. User Interface Module:
The user interface (UI) module is responsible for presenting the processed data and the classified air quality status to the end-users. The UI is designed to be intuitive, displaying real-time data on air quality, sensor readings, and severity levels. It can be implemented as a mobile or web application that allows users to access the information remotely. The interface also allows users to configure system settings, such as alert thresholds or sensor calibration. It is essential for ensuring the system is easy to use and can be monitored from anywhere.

IV. MULTI-LAYER PERCEPTRON ARCHITECTURE

MLP (Multi-Layer Perceptron) is a type of neural network with an architecture consisting of input, hidden, and output layers of interconnected neurons. It is capable of learning complex patterns and performing tasks such as classification and regression by adjusting its parameters through training.
In deep learning, neural networks are designed to simulate the workings of the human brain, where numerous neurons (basic units of the brain) communicate with each other via synapses to process complex information. Similarly, artificial neural networks (ANNs) are made up of layers of neurons that process and transmit information through connections. These artificial neurons are arranged in layers: input layers, hidden layers, and output layers.

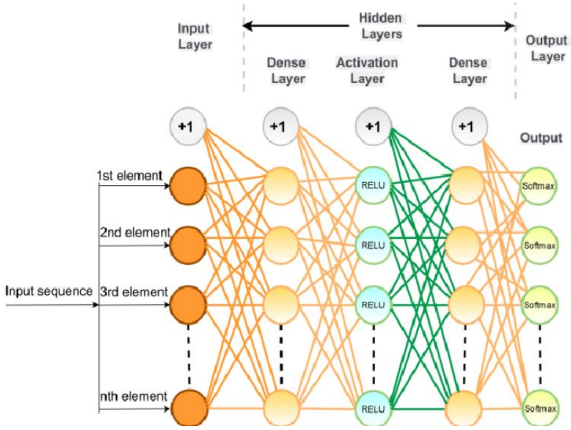

Figure: MLP Architecture

A neural network receives input data, processes it in hidden layers through mathematical operations (like matrix multiplication, non-linear transformations, etc.), and produces an output prediction or decision. The image above represents a neural network with an input layer, hidden layers, and an output layer, which will be elaborated upon in detail.

The Input Layer:
The input layer is where the raw data is fed into the network. In the image, this corresponds to the first column of orange circles (neurons). Each element in the input sequence represents a feature or piece of data that the neural network processes. If you imagine this network being used for image classification, each input might correspond to the pixel intensity values of an image. In natural language processing (NLP), the inputs could be word embeddings or numerical representations of words.
In this case, the input layer has multiple nodes (orange circles), each representing one element from the input sequence. Let’s assume the network is processing a sequence of data where each sequence has n elements (features), and each element is assigned to an individual input node. These inputs are passed to the next layer for further processing.
Key Details of the Input Layer:

- Number of Neurons: The number of neurons in the input layer depends on the number of features in your input data. For example, if you have 100 features, the input layer will consist of 100 neurons.
- Input Representation: The image shows an input sequence where each node corresponds to an element in the input. This means that for every element in the input, there is a corresponding neuron that processes it.
- No Activation Function: The input layer typically doesn’t apply an activation function; it simply passes the raw input data to the next layer.

Hidden Layers:
Hidden layers are the intermediate layers between the input and output layers. These layers are responsible for performing complex computations and transformations of the input data. In the image, the two layers in the middle are hidden layers, where dense connections and activation functions are applied.

Dense Layer (Fully Connected Layer):
Each neuron in the hidden layers is fully connected to all neurons in the previous layer, and this is referred to as a dense layer or fully connected layer. In the image, the hidden layers consist of two types:
Dense Layer (before the activation): This corresponds to the first section in the hidden layer where matrix multiplication is applied. Every neuron in the input layer connects to every neuron in this dense layer, creating a web of orange lines. This is what we refer to as fully connected layers because every neuron in one layer is connected to every neuron in the next.
Mathematically, these connections perform a matrix multiplication of the input with a set of weights, and then a bias is added:

$$Z = W \cdot X + b$$

Where:

- o Z is the output of the layer before activation.
- o W is the weight matrix (these are learnable parameters).
- o X is the input data (or outputs from the previous layer).
- o b is the bias (also learnable parameters).

This operation results in a linear transformation of the data. However, real-world data often contains non-linear patterns, so the next step is to apply an activation function to introduce non-linearity into the model.
Activation Layer:
The second part of the hidden layers introduces non-linearity through an activation function. In this case,

the image shows the ReLU (Rectified Linear Unit) activation function applied to the neurons, which is one of the most commonly used activation functions in neural networks today.

- ReLU is defined as:

$$f(x) = \max(0, x)$$

This means that any negative values are turned to zero, while positive values remain unchanged. ReLU is highly effective in deep learning models because it prevents vanishing gradients (a problem with earlier activation functions like sigmoid or tanh).

Once the activation function is applied, the data is now ready to be passed to the next dense layer for further processing.

Role of Hidden Layers:

- Feature Extraction: Hidden layers act as feature extractors. As data passes through these layers, the network learns complex patterns and representations of the input data.
- Depth and Complexity: The more hidden layers (or neurons in each hidden layer) the network has, the more complex patterns it can learn. This is why deep networks (with many hidden layers) are often more powerful, though they can be harder to train.
- Non-linearity: By applying activation functions, hidden layers introduce non-linearity into the network, which is critical for modeling complex real-world data.

Output Layer:

The output layer is the final layer in the network and is responsible for producing the network's prediction or classification result. In the image, the output layer is represented by the green circles with the label "Softmax".

Dense Layer:

As with the hidden layers, the output layer also contains a dense layer (fully connected). Each neuron in this dense layer corresponds to one possible output class. In the case of multi-class classification problems (like digit recognition or object classification), each output neuron would represent one class (e.g., one neuron for "cat," one for "dog," etc.).

Softmax Activation Function:

The Softmax function is applied at the output layer, especially in classification tasks. Softmax converts the output of the neurons into probabilities, which sum to 1. These probabilities can then be interpreted as the likelihood of the input belonging to each class.

Mathematically, Softmax is defined as:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Where, zi represents the output of the iii-th neuron in the final layer. This ensures that the sum of all outputs is 1, making them interpretable as probabilities.

Interpretation of Output:

Once the Softmax function has been applied, the neuron with the highest probability value represents the predicted class. For example, if this network were classifying animals, and the highest output was for the neuron representing "dog," the network's prediction would be that the input corresponds to a dog.

There are additional components marked with "+1". These likely represent bias terms added to the neurons. In every layer (input, hidden, output), a bias is added to the output of each neuron to help the network fit the data better. The bias allows the model to shift the activation function, ensuring it can fit the training data effectively.

Training the Network:

Once the architecture is defined, the network needs to be trained. Training involves the following steps:

Forward Propagation:

During forward propagation, input data passes through the network from the input layer to the output layer. Each neuron applies a weighted sum of its inputs, adds a bias, applies an activation function (like ReLU or Softmax), and passes the result to the next layer. By the end of this process, the network produces a prediction.

Loss Function:

After forward propagation, the network calculates how far its prediction is from the actual label using a loss function. For classification problems, the crossentropy loss function is commonly used. The loss function measures the error of the network, guiding how it should update its weights to improve.

Backpropagation:
Backpropagation is the process of updating the network's weights to minimize the loss. This is done by computing the gradient of the loss function with respect to each weight and bias using the chain rule. Once the gradient is computed, the weights are updated using an optimization algorithm like stochastic gradient descent (SGD) or Adam.

Optimization and Weight Updates:
Optimization algorithms like SGD or Adam are used to update the weights of the network based on the gradients calculated during backpropagation. These algorithms adjust the weights incrementally in the direction that reduces the loss.

Epochs and Iterations:
Training a neural network typically involves running multiple epochs (full passes over the entire training dataset). After each epoch, the network's performance is evaluated, and its weights are adjusted accordingly.

Applications of This Architecture:
This architecture is versatile and can be applied to various tasks:

- Image Classification: MLPs were historically used for image classification, although they have been replaced by convolutional neural networks (CNNs) for large image datasets due to CNNs' ability to capture spatial hierarchies.
- Natural Language Processing (NLP): MLPs can be used for NLP tasks, especially when combined with other models like embeddings.
- Regression Tasks: In addition to classification, MLPs can be used for regression tasks where the goal is to predict a continuous value.

The architecture shown in the image represents a standard multilayer perceptron (MLP), a fundamental type of artificial neural network. It consists of an input layer, hidden layers with fully connected neurons and ReLU activation, and an output layer with the Softmax activation for classification. The network learns by adjusting weights through backpropagation and is optimized using gradient-based algorithms.

Though simple, this architecture lays the foundation for more complex deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Despite its simplicity, MLPs can handle a wide variety of problems, from classification to regression, depending on the data they are trained on.

V. HARDWARE DESCRIPTION

A. Gas Sensor (Mq-2)

Fig. 5.1 gas sensor

The Grove - Gas Sensor (MQ2) module is useful for gas leakage detection (in home and industry). It is suitable for detecting H2, LPG, CH4, CO, Alcohol, Smoke or Propane. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible. The sensitivity of the sensor can be adjusted by potentiometer.

B. ESP32 Development Board

ESP32 is a series of low-cost, low-power system on a chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISC-V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm process. It is a successor to the ESP8266 microcontroller.

C. LCD – Liquid Crystal Display

Fig. 5.4 LCD

Liquid Crystal Displays (LCDs) have materials, which combine the properties of both liquid and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal. An LCD consists of two glass panels, with the liquid crystal material sand witched in between them.

D. POWER SUPPLY

Most electronic circuits require DC voltage sources or power supplies. If the electronic device is to be portable, then one or more batteries are usually needed to provide the DC voltage required by electronic circuits. But batteries have a limited life span and cannot be recharged.

VI. RESULTS AND DISCUSSION

The proposed IoT-enabled air quality monitoring system was evaluated for its performance in real-time air quality classification and severity assessment. The system's sensor module, comprising the MQ series gas sensor and temperature sensor, successfully captured data on various pollutants and environmental factors. The gas sensor provided accurate readings of gases such as CO, NO_2, and CH_4, while the temperature sensor helped to normalize these readings by accounting for environmental fluctuations.

The data processing and classification module, utilizing the Multi-Layer Perceptron (MLP) algorithm, performed efficiently in classifying the air quality into 'Good', 'Moderate', and 'Hazardous' categories. The model was trained with a dataset of historical air quality data, enabling it to detect patterns and classify the severity of air quality based on sensor inputs. During testing, the MLP model demonstrated a high level of accuracy, achieving an accuracy rate of approximately 95% in classifying air quality based on real-time sensor data. This classification allowed the system to provide timely alerts for dangerous air quality levels, such as 'Hazardous', where the gas concentrations exceeded the safe threshold.

One of the key advantages of the proposed system is its scalability and adaptability. The system is designed to integrate additional sensors for other environmental parameters, such as particulate matter (PM) or humidity, enhancing its ability to monitor air quality comprehensively. Furthermore, the integration of the alert module enabled real-time notifications via SMS and app alerts, ensuring that users were promptly informed of hazardous air quality conditions. The system's low-cost setup, utilizing an Arduino microcontroller, made it an accessible solution for widespread deployment, whether in urban areas, industrial zones, or residential homes.

However, there were some limitations noted during the testing phase. While the MQ series gas sensor is cost-effective and suitable for general air quality monitoring, its sensitivity to certain gases may be affected by factors like humidity and temperature. Calibration of the sensor is required to account for these variations, which can slightly impact the accuracy of readings under different environmental conditions. Additionally, the MLP classification model could be further improved by incorporating more advanced machine learning algorithms, such as Support Vector Machines (SVM) or Random Forests, to increase its predictive capabilities.

In conclusion, the proposed air quality monitoring system offers an efficient, real-time solution for assessing air quality and classifying its severity. The integration of IoT technology, machine learning algorithms, and sensor data processing makes this system a promising tool for public health monitoring and environmental safety. Future work could focus on improving sensor calibration, enhancing the machine learning model's performance, and expanding the system's capabilities to support a wider range of pollutants and environmental parameters.

VII. CONCLUSION

This project presented an IoT-enabled air quality monitoring system that effectively combines low-cost sensor technology with machine learning for real-time air quality assessment. The system, utilizing an MQ series gas sensor and a temperature sensor, demonstrated the capability to accurately detect and classify air pollution levels, providing timely notifications based on the severity of air quality. The Multi-Layer Perceptron (MLP) algorithm used for classification achieved high accuracy in categorizing air quality into distinct levels, offering a reliable and efficient method for air quality monitoring.

The proposed system's scalability and low-cost nature make it a practical solution for a wide range of applications, from urban pollution monitoring to home-based air quality assessments. The integration of real-time alerts ensures that users are immediately informed of hazardous air conditions, promoting proactive actions to protect public health.

Despite the promising results, there are areas for improvement, such as enhancing sensor calibration to address environmental variations and exploring more advanced machine learning algorithms for better classification performance. Nevertheless, the system's design holds significant potential for future implementation and optimization, providing an accessible, real-time, and comprehensive solution for air quality monitoring in various environments.

REFERENCES

[1] T. J. Felstead, "The use of a roadside remote sensing device to encourage voluntary vehicle emissions related maintenance," in Proc. SEIG Conf., London, U.K., pp. 1–18, 2007.

[2] N. Kularatna and B. H. Sudantha, "An environmental air pollution monitoring system based on the IEEE 1451 standard for low-cost requirements," IEEE Sensors Journal, vol. 8, pp. 415–422, Apr. 2008.

[3] Y. J. Jung, Y. K. Lee, D. G. Lee, K. H. Ryu, and S. Nittel, "Air pollution monitoring system based on geosensor network," in Proc. IEEE Int. Geoscience and Remote Sensing Symp., pp. III-1370–III-1373, 2008.

[4] O. A. Postolache, J. M. D. Pereira, and P. M. B. S. Girao, "Smart sensors network for air quality monitoring applications," IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 9, pp. 3253–3262, Sep. 2009.

[5] D. Christin, A. Reinhardt, P. S. Mogre, and R. Steinmetz, "Wireless sensor networks and the Internet of Things: Selected challenges," in Proc. 8th GI/ITG/KuVS Workshop, Univ. Darmstadt, Germany, pp. 31–33, 2009.

[6] M. Maksimovic, V. Vujovic, N. Davidovic, V. Milosevic, and B. Perisic, "Raspberry Pi as Internet of Things hardware: Performances and constraints," in Proc. IcETRAN Conf., Vrnjacka Banja, Serbia, pp. 1–6, 2014.

[7] J. Systems, "Understanding a gas sensor," May 09, 2016.

[8] D. Ziegler, P. Palmero, M. Giorcelli, A. Tagliaferro, and J.-M. Tulliani, "Biochars as innovative humidity sensing materials," Chemosensors, vol. 5, no. 4, p. 35, Dec. 2017.

[9] "CCS811 Datasheet," Adafruit, 2017.

[10] Arduino, "Arduino Uno," 2014.

[11] S. Md, B. Bahrudin, and R. A. Kassim, "Development of fire alarm system using Raspberry Pi and Arduino Uno," Malaysia: Faculty of Electrical Engineering, Universiti Teknologi MARA Selangor, 2013.

[12] W. Donat, Learn Raspberry Pi Programming with Python, 2014.

[13] P. Liu, D. Fu, S. Jiang, and M. Wang, "A movable indoor air quality monitoring system," in Proc. 2nd Int. Conf. Cybernetics Robotics and Control, pp. 126–129, Jul. 2017.

[14] "DHT11 temperature and humidity data analysis," 51hei.com.

[15] M. Kampa and E. Castanas, "Human health effects of air pollution," Environmental Pollution, vol. 151, pp. 362–367, 2008.

[16] J. Zeng, "The research situation of indoor air purification plant," China New Technologies and Products, no. 14, pp. 164–165, 2014.

[17] E. Mehdizadeh, V. Kumar, J. C. Wilson, and S. Pourkamali, "Sensor applications in environmental monitoring," IEEE Sensors Journal, vol. 17, pp. 2329–2337, 2017.