

**Real Time Classification of Smoke using IoT**

UG PROJECT

**MEMBERS**

Akshit Singla - 20095004

Hemant Joshi - 20095044

Vedanth Powar - 20095121

Prashasti Tripathi - 20095083

**Under the supervision of:**

Dr. Navin Singh Rajput

**(DEPARTMENT OF ELECTRONICS ENGINEERING)**

**(INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI)**

**CERTIFICATE**

This is to certify that the UG Project entitled “Real Time Classification of Smoke using IoT” submitted by Mr. Akshit Singla (20095004), Mr. Hemant Joshi (20095044), Ms. Prashasti Tripathi (20095083) and Mr. Vedanth Powar (20095121), to the Department of Electronics Engineering, Indian Institute of Technology (Banaras Hindu University) Varanasi, in partial fulfilment of the requirements for the award of the degree “Bachelor of Technology” in Electronics Engineering is an authentic work carried out at Department of Electronics Engineering, Indian Institute of Technology (Banaras Hindu University) Varanasi under my supervision and guidance on the concept vide project grant as acknowledged.

Supervisor Signature

**DECLARATION**

I hereby declare that the work presented in this project titled “**Real Time Classification of Smoke using IoT**” is an authentic record of our own work carried out at the Department of Electronics Engineering, Indian Institute of Technology (Banaras Hindu University), Varanasi as requirement for the award of degree of Bachelors of Technology in Electronics Engineering, submitted in the Indian Institute of Technology (Banaras Hindu University) Varanasi under the supervision of Dr. Navin Singh Rajput, Department of Electronics Engineering, Indian Institute of Technology (Banaras Hindu University) Varanasi. It does not contain any part of the work, which has been submitted for the award of any degree either in this Institute or in other University/Deemed University without proper citation.

Akshit Singla Vedanth Powar

Hemant Joshi Prashasti Tripathi

**ABSTRACT**

This project presents a comprehensive Internet of Things (IoT) solution for gas monitoring and prediction. Utilizing a network of MQ series gas sensors interfaced with an ESP8266 microprocessor, the system collects real-time data on gas concentrations. The data is transmitted to MongoDB, triggering AWS Lambda functions for seamless upload to an S3 bucket. An EC2 instance hosts a machine learning model trained to predict the presence of specific gases, enhancing the system's predictive capabilities. Challenges such as sensor drift, data integrity, and security are addressed, ensuring the robustness of the overall gas monitoring infrastructure. The project showcases the integration of hardware, cloud services, and machine learning for scalable, accurate, and real-time gas monitoring.

**Content**

[Abstract](#_TOC_250008) 4

1. Introduction
   1. [Overview…………………………………………………………………...](#_TOC_250007)6
   2. [Objectives………………………….…………………………….….](#_TOC_250006).6
2. Literature Review of Existing Techniques and Devices 7
   1. Existing Techniques…….……...………………………………………..7
   2. Problems with Existing Devices...……………………………………………......9
3. Work Done 10
   1. Hardware…………………………………….………………………...10
   2. Data Flow……………………………………………………………………...13
   3. [Cloud](#_TOC_250005) Services Used……………………………………………………...14
   4. Dataset………….....…………………………….………………………...15
   5. [Developing and Selecting a Machine Learning Model…………………](#_TOC_250004) ...16
4. Experiments and Results 18
   1. [Score Tables](#_TOC_250002) 18
5. Challenges, Conclusions & Future Scope 20
   1. Challenges and Solutions……………………………...……………….20
   2. Conclusions.…………………………………………………………………...21
   3. Future Scope……….……………………………………………………...21
6. Bibliography 22

Chapter 1

**INTRODUCTION**

* 1.1 Overview

This project endeavors to create an advanced Internet of Things (IoT) system for real-time gas monitoring. The system employs MQ series gas sensors (MQ2, MQ3, MQ4, MQ6, MQ7, MQ8, MQ9, MQ135) interfaced with an ESP8266 microprocessor. The collected sensor data is stored in MongoDB and triggers AWS Lambda for seamless data upload to an S3 bucket. An EC2 instance hosts a website and a machine learning (ML) model trained to predict the presence of specific gases. The ML model is capable of identifying four distinct classes: Burning Wire, Camphor, Air, and Incense Stick. Then this data is displayed on our front-endApplication.

* 1.2 Objectives
* Implement a robust IoT system for continuous gas monitoring in diverse environments.
* Employ MQ series gas sensors for accurate and real-time data acquisition.
* Establish a scalable data storage solution using MongoDB.
* Utilize AWS services (Lambda, S3) for automated data transfer and storage.
* Develop and deploy an ML model on an EC2 instance capable of predicting the presence of four gases.
* Integrate prediction results into a user-friendly website.

Chapter 2

**Literature Review of Existing Techniques**

* 2.1 Existing Techniques

Gas detectors are devices used to detect and monitor the presence of gases in an environment. There are various types of gas detectors and techniques used to detect gases, and they are used in a variety of applications, including industrial safety, environmental monitoring, and medical applications. In this literature survey, we will review some of the common gas detectors and techniques.

**Catalytic Combustion Detectors (CCD):**

Catalytic combustion detectors (CCD) are commonly used to detect combustible gases, such as methane and propane. These detectors work by oxidizing the gas in a catalytic sensor, producing a measurable change in resistance that can be used to trigger an alarm. CCDs are commonly used in industrial settings, such as refineries and chemical plants.

**Infrared (IR) Gas Detectors:**

Infrared (IR) gas detectors are used to detect a wide range of gases, including hydrocarbons, carbon dioxide, and refrigerants. These detectors work by measuring the absorption of IR radiation by the gas being detected. IR detectors are commonly used in HVAC systems, refrigeration, and other industrial applications.

**Electrochemical Gas Sensors:**

Electrochemical gas sensors are used to detect toxic gases, such as carbon monoxide, chlorine, and hydrogen sulfide. These sensors work by measuring the electrical current generated when a gas reacts with an electrode. Electrochemical sensors are commonly used in industrial settings and in personal gas monitors.

**Photoionization Detectors (PID):**

Photoionization detectors (PID) are used to detect volatile organic compounds (VOCs), such as benzene and toluene. These detectors work by ionizing gas molecules with ultraviolet light, producing a measurable current that can be used to trigger an alarm. PIDs are commonly used in environmental monitoring and industrial settings.

**Metal Oxide Semiconductor (MOS) Gas Sensors:**

Metal oxide semiconductor (MOS) gas sensors are used to detect a wide range of gases, including carbon monoxide, nitrogen dioxide, and hydrogen. These sensors work by measuring the change in resistance when a gas reacts with a metal oxide film. MOS sensors are commonly used in industrial and residential settings.

**Fourier Transform Infrared (FTIR) Spectroscopy:**

Fourier transform infrared (FTIR) spectroscopy is a technique used to detect and identify gases. FTIR works by measuring the absorption of IR radiation by a gas sample, producing a unique spectral fingerprint that can be used to identify the gas. FTIR is commonly used in environmental monitoring and industrial settings.

**Laser-Based Gas Sensors:**

Laser-based gas sensors use laser light to detect and measure the concentration of gases. These sensors work by measuring the absorption of laser light by a gas sample, producing a measurable change in light intensity that can be used to trigger an alarm. Laser-based sensors are commonly used in environmental monitoring and industrial settings.

In conclusion, there are various types of gas detectors and techniques used to detect and monitor the presence of gases in an environment. The choice of detector and technique will depend on the specific gas being detected and the application.

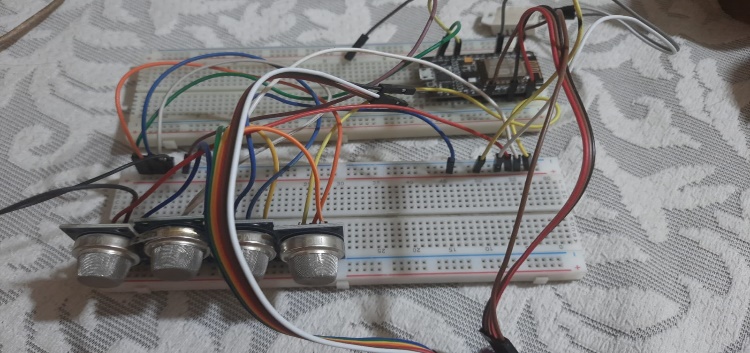
* 2.2 Problems with existing devices
* **Limited functionality:** Many existing devices on the market have limited functionality, and are only capable of measuring a few key metrics related to air quality. This means that they may not provide a comprehensive picture of air quality in a given environment.
* **High cost:** Many existing air quality monitoring devices are quite expensive, which can make them prohibitive for some users. This may limit their accessibility, particularly in developing countries or low-income areas where air quality may be a significant concern.
* **Lack of connectivity:** Some existing devices are not equipped with the necessary connectivity features to transmit data to the cloud in real time. This can make it difficult to monitor air quality levels in real time or to track changes over time.
* **Difficult to install:** Some air quality monitoring devices can be difficult to install or require significant expertise to set up properly. This can be a major barrier for many users, particularly in smaller organizations or homes where there may not be dedicated technical staff available.

Chapter 3

**Work Done**

* 3.1 Hardware Setup

The hardware setup includes MQ series gas sensors connected to an ESP8266 microprocessor. Each sensor is strategically placed to capture a comprehensive range of gas concentrations. Initially 4 sensors were used to detect gases/ smoke but finally 8 gas sensors were used because of not so high accuracy of previous classifier models.





The hardware devices used are: -

* **Bread Board:** A breadboard is a device used to build and test electronic circuits without the need for soldering. It consists of a plastic board with a grid of holes arranged in rows and columns. The holes are connected together by metal clips that allow electronic components, such as resistors, capacitors, and integrated circuits, to be easily inserted and connected together to build a circuit.
* **Jumper Wires**: Jumper wires are electrical wires with a connector pin at each end used to connect electronic components on a breadboard or other circuit board. They can also be used to connect components on separate breadboards or to connect a breadboard to an external device, such as a microcontroller or sensor.
* **ESP 8266 Microcontroller:** The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capabilities, produced by Espressif Systems. It is commonly used in Internet of Things (IoT) devices due to its small size, low power consumption, and built-in Wi-Fi connectivity. The ESP8266 microprocessor has 12 General Purpose Input Output (GPIO) pins, which can be used for a variety of purposes, including controlling peripherals such as sensors and actuators. It also has two serial communication pins (TX and RX), which can be used to communicate with other devices, such as a computer or another microcontroller.

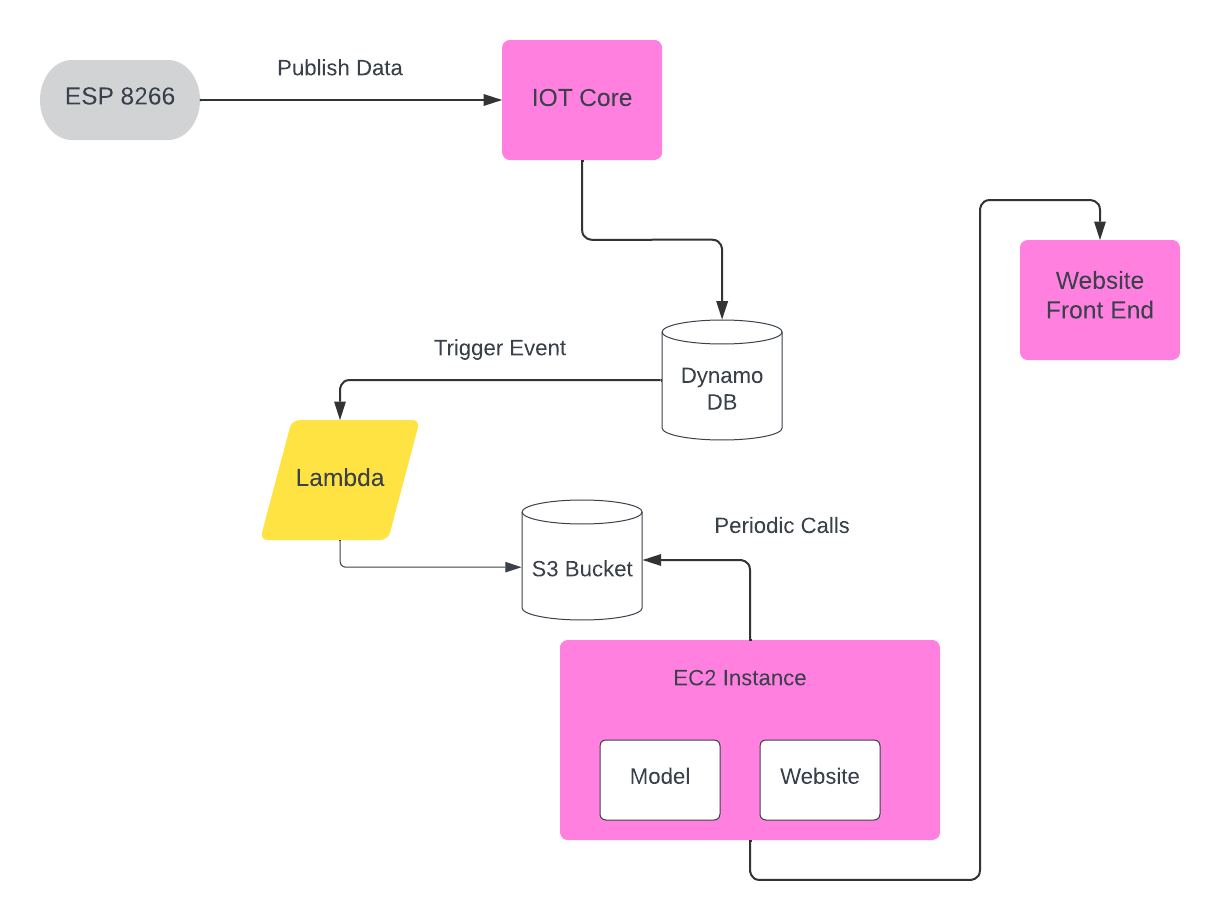
In addition, the ESP8266 has an onboard Wi-Fi module, which allows it to connect to a Wi-Fi network and communicate with other devices over the internet. It also has a built-in flash memory for storing code and data, as well as a Real-Time Clock (RTC) for keeping track of time even when the device is powered off. The ESP8266 can be programmed using the Arduino IDE, as well as other programming languages such as MicroPython and Lua.

* **Gas Sensors**: These sensors use a small heater and a sensing element to detect the presence of various gases in the air. The sensing element changes its electrical resistance when exposed to the target gas, and this change in resistance is then measured and used to calculate the gas concentration. Different sensors used are: -

1. **MQ2 Gas Sensor:** The MQ2 gas sensor is commonly used to detect smoke, methane, propane, butane, and other gases. It is cost-effective, easy to use, and has a high sensitivity to a wide range of gases.
2. **MQ3 Alcohol Sensor:** The MQ3 alcohol sensor is used to detect alcohol vapor in the air. It is commonly used in breathalyzer devices, alcohol detectors, and in the automotive industry to detect alcohol levels in the driver's breath.
3. **MQ4 Methane Sensor:** The MQ4 methane sensor is used to detect methane gas in the air. It is commonly used in gas leak detectors, gas monitoring systems, and in the oil and gas industry.
4. **MQ8 Hydrogen Sensor:** The MQ8 hydrogen sensor is used to detect hydrogen gas in the air. It is commonly used in gas leak detectors, fuel cell monitoring systems, and in the automotive industry.
5. **MQ6 LPG Sensor:** The MQ6 LPG sensor is used to detect LPG gas in the air. It is commonly used in gas leak detectors, gas monitoring systems, and in the oil and gas industry.
6. **MQ7 Carbon Monoxide Sensor:** The MQ7 carbon monoxide sensor is used to detect carbon monoxide gas in the air. It is commonly used in the automotive industry.
7. **MQ9 Carbon Monoxide and Flammable Gas Sensor:** The MQ9 carbon monoxide and flammable gas sensor is used to detect carbon monoxide and flammable gases such as propane and methane in the air. It is commonly used in gas leak detectors, gas monitoring systems, and in the automotive industry.
8. **MQ135 Air Quality Sensor:** The MQ135 air quality sensor is used to detect a wide range of gases, including carbon dioxide, ammonia, and benzene. It is commonly used in air quality monitors & environmental monitoring systems.

Overall, the benefits of using these gas sensors include their cost-effectiveness, ease of use, and high sensitivity to specific gases or environmental factors. They are essential tools in a variety of applications, including environmental monitoring, industrial safety, and home automation. Initially We used 4 sensors to collect and classify data but due to its low accuracy we used 8 sensors.

* 3.2 Data Flow
* Data Acquisition: Gas sensor data is continuously collected by the ESP8266 microprocessor.
* Event Trigger: MongoDB triggers an event upon receiving new sensor data, invoking a Lambda function.
* Data Upload: Lambda function uploads the data to an S3 bucket for efficient and scalable storage.
* Machine Learning Prediction: An EC2 instance regularly retrieves the latest data from the S3 bucket.
* The ML model processes the last 10 rows of data and predicts the presence of Burning Wire, Camphor, Air, or Incense Stick for the most recent entry.
* Result Display: Prediction results are sent to the website for display, providing real-time information to users.



* 3.3 Cloud Services Used
* MongoDB:

MongoDB is a NoSQL database that provides a flexible, scalable, and high-performance solution for storing and managing data. Its document-oriented structure allows for easy retrieval and manipulation of data, making it ideal for diverse applications, including IoT systems like our gas monitoring project.

* AWS Lambda:

AWS Lambda is a serverless computing service that executes code in response to events, eliminating the need for server management. It enables seamless integration in our project by triggering functions in response to MongoDB events, facilitating automated data transfer to the S3 bucket.

* Amazon IoT Core:

Amazon IoT Core is a managed cloud service that enables secure and scalable communication between IoT devices and the cloud. It plays a crucial role in our gas monitoring system, ensuring secure data transmission from sensors to MongoDB, contributing to the overall reliability of the IoT infrastructure.

* AWS EC2:

Amazon Elastic Compute Cloud (EC2) is a scalable virtual server in the cloud, providing on-demand computing resources. In our project, EC2 hosts the website and the machine learning model, offering flexibility and scalability to handle data processing, predictions, and user interface functionalities.

* S3 Bucket:

Amazon S3 (Simple Storage Service) is a scalable object storage solution in AWS. It serves as a central repository for our gas sensor data, facilitating seamless storage and retrieval. Its durability, security features, and easy integration with other AWS services make it an essential component in our IoT architecture.

* 3.4 Dataset

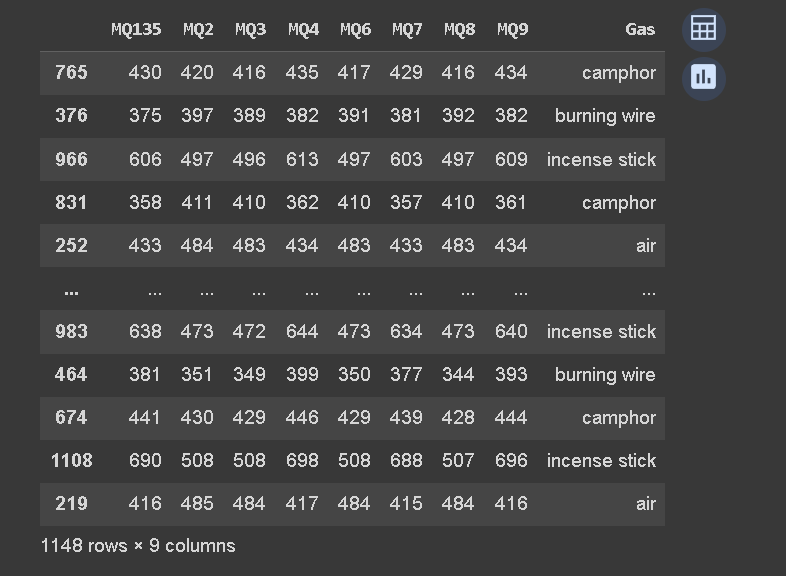
Dataset was collected manually in the lab. There was use of 4 different gases –

* 1. Camphor
  2. Electric Wire
  3. Incense stick
  4. Air

Some of the features of this dataset are:

* Classification Data
* Instances: ~4000 (1000 each)
* Attributes: 8
* Needs Pre-Processing – Min-Max Scaling.

Sample Dataset:



* 3.5 Developing and Selecting a Machine Learning Model

Data was collected from the ESP8266 sensors and sent to AWS IoT Core.

Lambda function was then used to save the sensor data to an S3 bucket

Then the data was cleaned and pre-processed using Python and Pandas.

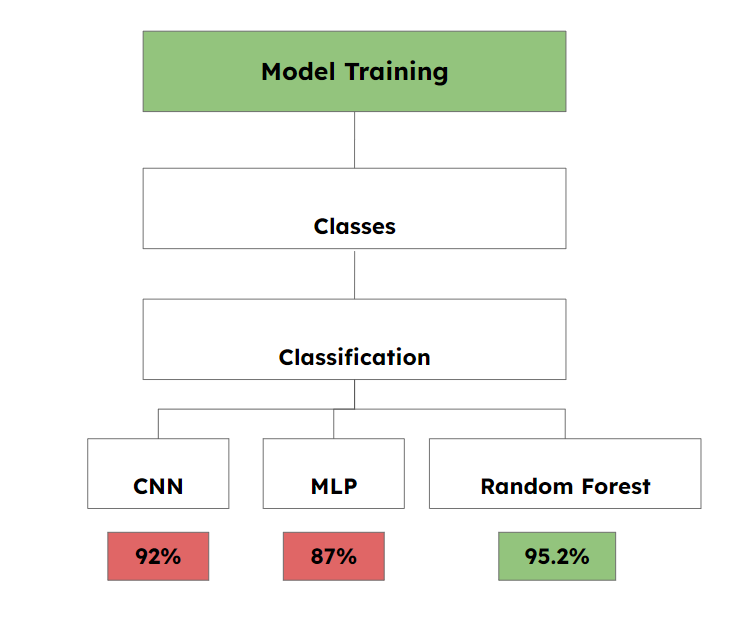
During pre-processing, min-max scaling was used, missing values were removed, Label encoding and One Hot encoding was used.

After pre-processing, the data was explored and visualized to gain insights and identify patterns.

With the insights from data exploration, several machine learning models were trained and developed using scikit-learn, including linear regression, decision trees, and random forests

The models were evaluated using various metrics such as mean squared error and accuracy, and selected the best performing model.

The selected model was then further fine-tuned by adjusting hyperparameters using techniques such as cross-validation and grid search and finally, the model was tested using a held-out dataset to verify its performance on unseen data.



**Models Used: -**

* **Multilayer Perceptron (MLP):**

The Multilayer Perceptron (MLP) is a feedforward neural network that consists of multiple layers of interconnected nodes (neurons). The input layer receives the input data, and the output layer produces the final output. The hidden layers are used to process the input data and extract relevant features. Each neuron in the network receives input from the previous layer, applies a non-linear activation function, and passes the output to the next layer. The weights of the connections between the neurons are learned through a process called backpropagation.

* **Convolutional Neural Network (CNN):**

The Convolutional Neural Network (CNN) is a type of neural network that is specifically designed for image processing tasks. It consists of multiple convolutional layers, which apply filters to the input image to extract features, and pooling layers, which down sample the feature maps to reduce their dimensionality. The output of the final convolutional layer is flattened and passed to one or more fully connected layers, which perform the final classification or regression task.

* **Random Forest Classifier:**

The Random Forest Classifier is an ensemble learning method that consists of multiple decision trees. Each decision tree is trained on a subset of the training data and selects a random subset of features to split the data at each node. The final prediction is made by taking the majority vote of the individual tree predictions.

Chapter 4

**Experiments and Results**

* 4.1 Score Tables:

One of the main advantages of MLP is that it can approximate any continuous function to arbitrary accuracy, provided that it has enough hidden layers and neurons. This makes it a powerful tool for solving complex problems in various fields. However, training an MLP can be computationally intensive and time-consuming, especially when dealing with large datasets.

|  |  |
| --- | --- |
| **MLP MODEL** | |
| Accuracy | 87% |
| Precision | 0.33 |
| Recall | 0.33 |
| F-score | 0.33 |

The random subset of features and training data used in each tree helps to reduce overfitting and improve the generalization ability of the model. The Random Forest algorithm can also handle missing data and maintain accuracy in the presence of noisy data.

|  |  |
| --- | --- |
| **RANDOM FOREST** | |
| Accuracy | 95.2% |
| Precision | 0.50 |
| Recall | 0.50 |
| F-score | 0.25 |

**CNN** stands for Convolutional Neural Network, which is a type of deep learning algorithm. CNNs can perform parallel processing on the input data, making them highly efficient for large-scale image classification tasks.

|  |  |
| --- | --- |
| **CNN** | |
| Train Accuracy | 92% |
| Test Accuracy | 95% |
| Test Loss | 0.139 |

Chapter 5

**Challenges, Conclusions & Future Scope**

* 5.1 Challenges and Solutions:
* **Sensor Drift:**

Challenge: Sensors may experience drift over time, leading to inaccuracies in gas concentration readings.

Solution: Implement regular calibration routines to correct sensor drift and maintain data accuracy.

* **Data Volume and Throughput:**

Challenge: As the number of sensors and data points increases, handling the volume and ensuring timely processing becomes challenging.

Solution: Optimize data processing algorithms and consider distributed computing solutions for improved throughput.

* **Machine Learning Model Accuracy:**

Challenge: The ML model's accuracy may be affected by changes in the environment and variations in gas concentrations.

Solution: Continuously update and retrain the model using new data to enhance accuracy and adapt to changing conditions.

* **Cost Optimization:**

Challenge: Cloud service costs can accumulate, especially with frequent data transfers and computation.

Solution: Implement cost-monitoring practices, leverage serverless architectures efficiently, and explore pricing models to optimize costs.

* **Adaptation to New Gas Types:**

Challenge: The system may need to adapt to detect and classify new gas types not initially considered.

Solution: Design the system to accommodate updates and additions to the gas classification model, ensuring scalability and versatility.

* 5.2 Conclusions:

The project successfully integrates hardware, cloud services, and machine learning to create an advanced gas monitoring and prediction system. The robust architecture ensures scalability, real-time data processing, and accurate predictions. We used classification models on the Disinfectants dataset like CNN, MLP and Random Forest and found Random Forest Classification Model giving highest accuracy with 95.2%.

* 5.3 Future Scope:

We wish to carry forward our task and finally create a cyber physical system with total integration of task. Some applications we aim for in near future:

* Real-time Alerts: Implement real-time alerts for abnormal gas levels.
* Model Refinement: Continuously refine the ML model for improved accuracy and expanded capabilities.
* User Interface Enhancements: Enhance the user interface of the website for a more intuitive user experience.
* Fit in exhaust pipes of vehicles to detect the vehicular emission and estimate the BS model of Vehicle and impose TAX accordingly.
* Put in Operation Theatres to monitor the amount and type of detergents used in OT.
* To check the quality of drive by analysing the pattern of gas emission during drive.
* Aim to create an app which can help in Vehicle Emission Monitoring, Pollution Monitoring and help in understanding their impact on Environment.

Chapter 6

**Bibliography**

[1] S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain, “DROCC: Deep robust one-class classification,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 3711–3721.

[2] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, “Efficient neural architecture search via parameters sharing,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 4095–4104.

[3] Elsts and R. McConville, “Are microcontrollers ready for deep learning-based human activity recognition?” Electronics, vol. 10, no. 21, p. 2640, Oct. 2021.

[4] S. Herath, H. Yan, and Y. Furukawa, “RoNIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2020, pp. 3146–3152.

[5] T. Hossain, M. Ahad, A. Rahman, and S. Inoue, “A method for sensorbased activity recognition in missing data scenario,” Sensors, vol. 20, no. 14, p. 3811, Jul. 2020.