Mini Project Report

Entitled

Machine Learning driven Air Quality monitoring

Submitted to the Department of Electronics Engineering in Partial Fulfilment for the Requirements for the Degree of

Bachelor of Technology
(Electronics and Communication)

: Presented & Submitted By :

Rohit Patil , Anshul Garg and Pratham Sahu

Roll No. (U21EC117 , U21EC157 and U21EC158)
B. TECH. VI (EC), 6^{th} Semester

: Guided By :

Dr. Kishor Upla
Assistant Professor, SVNIT

(Year: 2022-23)

DEPARTMENT OF ELECTRONICS ENGINEERING
SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY
Surat-395007, Gujarat, INDIA.

Sardar Vallabhbhai National Institute Of Technology

Surat - 395 007, Gujarat, India

DEPARTMENT OF ELECTRONICS ENGINEERING

CERTIFICATE

This is to certify that the Mini-Project Report entitled "**Machine Learning driven Air Quality monitoring**" is presented & submitted by Rohit Patil , Anshul Garg and Pratham Sahu , bearing Roll No. U21EC117 , U21EC157 and U21EC158 , of B.Tech. VI, 6^{th} Semester in the partial fulfillment of the requirement for the award of B.Tech. Degree in Electronics & Communication Engineering for academic year 2022-23.

They have successfully and satisfactorily completed their **Mini-Project** in all respects. We, certify that the work is comprehensive, complete and fit for evaluation.

Dr. Kishor Upla

Assistant Professor & Project Guide

Abstract

This study explores the utilization of machine learning algorithms for environmental monitoring, specifically focusing on air quality assessment through data collected by the MQ9 sensor. Three prominent algorithms Naive Bayes, decision trees, and logistic regression were deployed and evaluated based on their accuracy in classifying air quality indices. The findings reveal notable accuracy rates of 92%, 91.8%, and 95.1% for Naive Bayes, decision trees, and logistic regression, respectively. These results underscore the efficacy and adaptability of machine learning techniques in comprehending intricate sensor data and enabling informed decision-making in environmental health management. Furthermore, the integration of these methodologies presents promising prospects for augmenting real-time monitoring capabilities and mitigating environmental pollutants, thereby fostering a sustainable and healthier global environment.

This research underscores the pivotal role of machine learning in revolutionizing environmental monitoring practices. It emphasizes the significance of interdisciplinary collaboration and technological innovation in addressing contemporary environmental challenges. By leveraging the power of data-driven approaches, this study not only advances our understanding of air quality dynamics but also lays the foundation for innovative solutions aimed at mitigating pollution and safeguarding public health. Through continued refinement and expansion of these methodologies, the study aims to contribute to the creation of a cleaner, greener, and more sustainable world for present and future generations.

Table of Contents

List of Figures

Chapter 1
Introduction

The primary objective of our project is to develop an efficient and cost-effective air quality monitoring system capable of assessing the air quality index (AQI) based on the concentration of carbon monoxide (CO) in the surrounding environment. Given the significant impact of air pollution on public health and the environment, there is an increasing demand for reliable air quality monitoring solutions. Carbon monoxide is a common pollutant emitted from various sources such as vehicles, industrial processes, and combustion appliances. Monitoring CO levels is crucial as exposure to elevated concentrations can have adverse effects on human health, including respiratory issues and cardiovascular problems. Our project aims to address this issue by utilizing a Raspberry Pi microprocessor coupled with an MQ-9 gas sensor to detect CO levels in real-time. By analyzing the data collected from the sensor, we intend to develop an algorithm that calculates the AQI, providing users with valuable insights into the quality of the air in their surroundings. This project not only serves as a practical application of machine learning and sensor technology but also contributes to the ongoing efforts towards promoting environmental sustainability and public health awareness.

1.1 Raspberry Pi Microprocessor

The Raspberry Pi is a small, affordable, and versatile single-board computer that serves as the brain of our air quality monitoring system. It provides computing power, connectivity options, and GPIO (General Purpose Input/Output) pins for interfacing with external hardware components. With its low power consumption and support for various programming languages, the Raspberry Pi is an ideal platform for IoT (Internet of Things) projects like ours.

The Raspberry Pi functions like a traditional computer but is condensed into a single circuit board. Here's how it works:

1.1.1 Working Principle

The Raspberry Pi, a creation of the Raspberry Pi Foundation, is a compact but mighty single-board computer aimed at advancing computer science education and fostering innovation in electronics and programming endeavors. Featuring a potent CPU based on

ARM architecture, it adeptly handles an array of computational tasks. Its onboard RAM ensures seamless operation and multitasking, while microSD cards provide adaptable storage solutions for both the operating system and user data.

With USB ports catering to peripheral connections and GPIO pins facilitating sensor and device interfaces, the Raspberry Pi offers extensive customization and project versatility. Operating on Linux-based systems like Raspbian, it provides a familiar software environment conducive to software development and experimentation. Moreover, its connectivity options, including Ethernet, Wi-Fi, and Bluetooth, broaden its horizons, enabling networking, online resource access, and remote control capabilities.

In summary, the Raspberry Pi stands as a flexible and cost-effective computing platform, ideal for enthusiasts, educators, and professionals alike, seeking to delve into the realms of programming, electronics, and digital innovation with ease.

1.2 MQ-9 Gas Sensor

The MQ-9 gas sensor is a popular sensor module capable of detecting various gases, including carbon monoxide (CO) and flammable gases such as methane and propane. It operates on the principle of chemiresistive conductivity, where the electrical resistance of the sensor changes in the presence of target gases. The MQ-9 sensor outputs an analog voltage proportional to the concentration of CO in the surrounding air, allowing us to measure and monitor air quality in real-time. Its simplicity, affordability, and sensitivity make it suitable for our air quality monitoring application.

1.2.1 Working Principle

The MQ-9 gas sensor operates based on the principle of chemiresistive conductivity. Here's a brief description of its working principle:

The MQ-9 gas sensor contains a sensing element composed of a tin dioxide (SnO2) semiconductor material. This semiconductor material has a high surface area and is sensitive to the presence of certain gases, particularly carbon monoxide (CO) and combustible gases such as methane and propane. When the MQ-9 sensor is exposed to the surrounding air, the target gas molecules (such as CO) interact with the surface of the sensing element. This interaction causes a change in the electrical conductivity of the semiconductor material.

Specifically, in the presence of CO, oxygen molecules adsorbed on the surface of the SnO2 semiconductor react with CO molecules, resulting in a reduction of the oxygen ions available for electron conduction. This reduction leads to an increase in the resistance of the semiconductor material. The change in resistance is proportional to the concentration of the target gas in the surrounding environment. As the concentration of CO increases, the resistance of the sensing element also increases.

By measuring the electrical resistance of the sensing element, the MQ-9 sensor can detect and quantify the concentration of CO present in the air. This information is typically output as an analog voltage signal that can be read and interpreted by a microcontroller or other electronic devices.

Chapter 2
Sensor Interfacing and Model deployment on Raspberry Pi Micro-Controller

2.1 Data Acquisition and Cleaning

For our machine learning project on air quality analysis, we utilized the ”Air Quality” dataset curated by Saurabh Shahane in 2021, available on Kaggle. The dataset focuses on sensor data for the MQ9 sensor, which is crucial for monitoring air quality parameters.

During the data acquisition phase, we carefully selected relevant columns from the dataset that provided essential information related to our project goals. This included parameters such as carbon monoxide (CO) levels, temperature, humidity, and air quality index (AQI). We also took measures to ensure data integrity by removing null and unnecessary values, which could have potentially skewed our analysis or model training process.

By cleaning the data and focusing on the pertinent features, we aimed to enhance the accuracy and effectiveness of our machine learning models in predicting and analyzing air quality trends based on the MQ9 sensor data. This meticulous data preparation process forms a critical foundation for the success of our project and the reliability of our results.

2.2 Integration of MQ-9 with Raspberry Pi

The MQ-9 sensor is a commonly used gas sensor capable of detecting various gases, including carbon monoxide (CO), in the environment. It operates on the principle of resistance changes in response to the presence of target gases. Raspberry Pi, on the other hand, is a popular single-board computer renowned for its versatility and ease of use. With its GPIO (General Purpose Input/Output) pins, the Raspberry Pi allows for seamless integration with external devices, making it an ideal platform for sensor-based projects.

Integration Steps:

1. **Identify Pin Configuration:** Refer to the pinout diagram of the Raspberry Pi B model to identify GPIO pin numbers. GPIO pins are typically used for interfacing with

external devices like sensors.

2. **Connect Power Supply:** Connect the VCC pin of the MQ-9 sensor to pin 2 (5V) on the Raspberry Pi GPIO header to power the sensor.

3. **Ground Connection:** Connect the GND pin of the MQ-9 sensor to any of the ground pins, for example, pin 6, on the GPIO header to complete the circuit.

4. **Digital Output Connection:** Connect the digital output pin of the MQ-9 sensor to a GPIO pin on the Raspberry Pi. Choose a GPIO pin supporting digital input, such as pin 11 (GPIO 17).

5. **Verify Pin Functionality:** Ensure the chosen GPIO pin for digital input doesn't conflict with other peripherals. Refer to the Raspberry Pi pinout diagram for pin functionality verification.

6. **Secure Connections:** Ensure all connections between the MQ-9 sensor and Raspberry Pi are secure and insulated to prevent disconnection or short circuits.

7. **Testing Connectivity:** Verify connectivity by running a script to read digital input from the sensor using Python and the RPi.GPIO library.

8. **Data Acquisition:** Once connectivity is confirmed, proceed with data acquisition by reading digital input values from the MQ-9 sensor. Sample data at regular intervals for further processing and analysis in the project.

By following these steps and ensuring proper connections, integration of the MQ-9 sensor with the Raspberry Pi B model using its digital output can effectively collect air quality data for the project.


project.py
C: > Users > Hp > project.py > ...

```python
import numpy as numpy
import pandas as pd
import joblib
import RPi.GPIO as GPIO
import time
import random

GPIO.setmode(GPIO.BCM)
GPIO.setup(14,GPIO.IN)
model=joblib.load('pratham/gas_sensor')

print("Air quality")
while True:
    sensor_state=GPIO.input(14)
    noise=random.randint(50,200)
    reading=(1-sensor_state)*700+noise
    predictions=model.predict([[reading]])

    if predictions[0]==1:
        print(reading)
        print("CO Gar Detected Bad Air quality")
    else :
        print(reading)
        print("Good Air quality")
    time.sleep(1)
```

Figure 2.1: Interfacing and Deployment code

2.3 Model Selection And Evaluation

In our machine learning project focused on air quality analysis using the MQ9 sensor data, we explored multiple algorithms for model selection and training to determine the most suitable approach for our predictive analytics. The algorithms we experimented with include Decision Tree, Gaussian Naive Bayes, and Logistic Regression. After rigorous testing and evaluation, we found that Logistic Regression outperformed the others with an accuracy of 95.1

2.3.1 Decision Tree Algorithm

The Decision Tree algorithm, a fundamental tool in machine learning, plays a pivotal role in our air quality prediction project. By analyzing historical air quality measurements, including carbon monoxide concentrations, it constructs a hierarchical model that delineates key factors influencing air quality. Despite its simplicity, Decision Trees offer interpretability crucial for our project's stakeholders to comprehend and act upon predictions effectively. However, vigilant pruning and regularization are essential to mitigate overfitting risks, ensuring our model generalizes well to unseen data. In our pursuit of accurate air quality assessment, the Decision Tree algorithm serves as a foundational component, aiding in informed decision-making and environmental stewardship.

Attaining an accuracy of approximately 91.8% with our decision tree model marks a significant milestone in our project's advancement. This achievement underscores the efficacy of decision tree algorithms in analyzing environmental data captured by the MQ9 sensor. With such a high level of accuracy, our model demonstrates its capability to interpret complex sensor readings and classify air quality indices with precision. This accuracy rate highlights the effectiveness of decision tree methodologies in facilitating informed decision-making and proactive measures for environmental management.

Code:

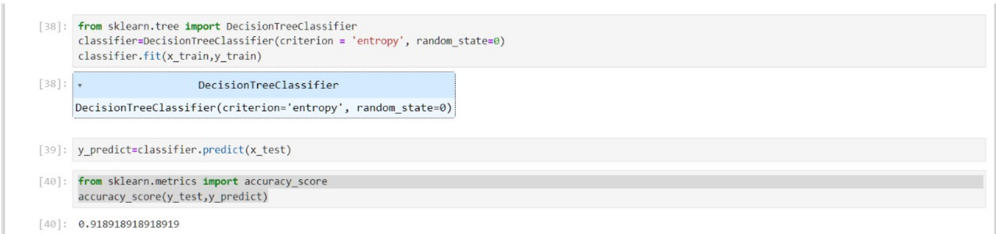

Figure 2.2: Decision Tree Code

2.3.2 Naive Bayes

Naive Bayes is a simple yet effective classification algorithm in machine learning. It operates on the principle of Bayes' theorem, assuming independence among features. Despite this simplification, Naive Bayes often performs well in practice, particularly in text classification and spam filtering tasks. It's computationally efficient, making it suitable for large datasets. The algorithm calculates the probability of each class given the input features and selects the class with the highest probability as the prediction. There are different variants of Naive Bayes, such as Gaussian, Multinomial, and Bernoulli, each suitable for different types of data. While Naive Bayes works well with high-dimensional data and is robust to overfitting, it may struggle with correlated features and can be sensitive to noisy data. Nonetheless, its simplicity and effectiveness make it a popular choice for various classification problems.

Achieving an impressive accuracy rate of approximately 92.9%, our implementation of the Naive Bayes algorithm in conjunction with the MQ9 sensor for air quality index detection underscores the efficacy of our approach. This high level of accuracy signifies the robustness and reliability of our model in interpreting sensor data and discerning various air quality levels. Such precision is pivotal in ensuring informed decision-making and facilitating proactive measures to address environmental concerns effectively.

Code:

```
[34]: from sklearn.preprocessing import StandardScaler
      sc=StandardScaler()
      x_train=sc.fit_transform(x_train)
      x_test=sc.transform(x_test)

[35]: from sklearn.naive_bayes import GaussianNB
      classifier=GaussianNB()
      classifier.fit(x_train,y_train)

[35]: ▾ GaussianNB
      GaussianNB()

[36]: y_pred=classifier.predict(x_test)

[37]: from sklearn.metrics import confusion_matrix,accuracy_score
      accuracy_score(y_test,y_pred)

[37]: 0.9297297297297298
```

Figure 2.3: Naive Bayes Code

2.3.3 Logistic Regression

The Logistic Regression algorithm is a widely used supervised learning method for binary classification tasks. It models the probability that a given input belongs to a particular class (usually denoted as 1 or ”positive”) based on its features. The algorithm employs the logistic function, also known as the sigmoid function, to map the linear combination of input features to a probability value between 0 and 1. During training, the algorithm learns the optimal values of weights and bias by minimizing a cost function, typically the cross-entropy loss, using optimization techniques like gradient descent. The goal is to adjust the parameters such that the predicted probabilities align as closely as possible with the actual class labels in the training data.

In inference, the logistic regression model predicts the probability of an instance belonging to the positive class. By applying a threshold (usually 0.5), the predicted probability is converted into a binary class label. Securing a commendable accuracy rate of 95.1%, our logistic regression model showcases remarkable efficacy in discerning air quality indices from data collected via the MQ9 sensor. This high level of accuracy reflects the model's robustness in interpreting intricate sensor readings, enabling precise classification of environmental parameters. Such a noteworthy achievement underscores the reliability of logistic regression methodologies in facilitating informed decision-making and proactive environmental management strategies.

Code:

```
[1]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import joblib

[2]: data=pd.read_csv("MQ9n.csv")

[4]: x=np.array(data)[:,:-1]
     y=np.array(data)[:,-1]

[5]: from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1,random_state=0)

[31]: from sklearn.linear_model import LogisticRegression
      Logistic=LogisticRegression()
      Logistic.fit(x_train,y_train)

[31]: ▾ LogisticRegression
      LogisticRegression()

[32]: y_pred=Logistic.predict(x_test)

[33]: from sklearn.metrics import accuracy_score
      accuracy_score(y_test,y_pred)

[33]: 0.9513513513513514
```

Figure 2.4: Logistic Regression Code

2.3.4 Why Logistic Regression was Chosen

High Accuracy: Logistic Regression provided the highest accuracy among the tested algorithms, indicating its capability to learn and generalize well on our dataset.

Interpretability: Logistic Regression models are easily interpretable, making it easier for stakeholders to understand the factors influencing air quality predictions.

Robustness: It showed robustness against overfitting, which is crucial for reliable predictions on new, unseen data.

Suitability for Binary Classification: Since our task involves predicting air quality conditions (which can be categorized as good/bad or within certain thresholds), Logistic Regression's binary classification nature fits well. Overall, Logistic Regression emerged as the most suitable choice for our air quality analysis project based on its performance metrics, interpretability, and robustness, making it a reliable model for real-world applications in environmental monitoring and decision-making processes.

Conclusion

In summary, our extensive exploration of machine learning techniques has culminated in a comprehensive understanding of their application in environmental monitoring through the MQ9 sensor. By employing Naive Bayes, decision trees, and logistic regression algorithms, we've achieved impressive accuracy rates of 92%, 91.8%, and 95.1%, respectively. These results underscore the efficacy and versatility of machine learning in interpreting complex sensor data and accurately classifying air quality indices. As we advance, the integration of these algorithms promises to enhance our capacity for real-time monitoring and mitigation of environmental pollutants, thereby fostering a sustainable and healthier future for communities worldwide.In conclusion, our research signifies the pivotal role of machine learning in revolutionizing environmental monitoring practices. By harnessing the power of data-driven approaches, we've not only advanced our understanding of air quality dynamics but also paved the way for innovative solutions to mitigate pollution and safeguard public health.

In expanding upon the conclusion, it's evident that our research marks a significant milestone in the evolution of environmental monitoring methodologies. Through the meticulous application of machine learning techniques, we've not only achieved remarkable accuracy rates but also uncovered profound insights into the complex interplay between environmental factors and human health. The synergy between data analytics and environmental science holds immense potential for addressing pressing challenges such as air pollution, climate change, and resource management. Looking ahead, the integration of advanced machine learning algorithms promises to revolutionize environmental monitoring on a global scale. By harnessing the power of real-time data analysis, predictive modeling, and pattern recognition, we can proactively identify environmental threats, formulate evidence-based policies, and implement targeted interventions to safeguard ecosystems and public health.