AI BASED AIR PURIFYING SYSTEM

PROJECT-21ECP302L

Submitted by

SUDARSH R M [RA2211004010038]

NARAYANAN P [RA2211004010052]

BALASAMINATHAN K [RA2211004010055]

Under the guidance of

Dr. A. Anilet Bala

(Assistant Professor, Department of Electronics & Communication Engineering)

BACHELOR OF TECHNOLOGY

in

ELECTRONICS & COMMUNICATION ENGINEERING

of

COLLEGE OF ENGINEERING AND TECHNOLOGY

S.R.M. NAGAR, Kattankulathur, Chengalpattu District

MAY 2025

ABSTRACT

Air pollution in urban areas has become a major public health concern, leading to respiratory issues and other severe diseases. Most current purification systems run constantly ignoring real-time air quality, which causes significant energy use and ineffectiveness. To overcome this problem the proposed solution incorporates an AI-integrated purification system. The system initiates monitoring of indoor air quality using a MQ-135 gas sensor interfaced with an Arduino microcontroller, calibrated for specific needs. Initial validation of the system was performed using Proteus, a simulation software after successful verification, the design was implemented in hardware and tested in a 3x4 meter room for practical air filtering operations. To enhance this performance, a dataset was created by integrating real-time air quality measurements with meteorological data obtained from the Open-Meteo weather forecast. This dataset was the basis for training a decision tree algorithm, which predicts future air quality values. The trained model was implemented on the ESP32 microcontroller, enabling the system to compare real-time sensor values with the predicted values and adjust the function accordingly. For the filtration process, a cabin air filter typically used in automobiles is combined with a fan to effectively remove pollutants from the air. The module sends live air quality data to mobile interface using the ESP32 Wi-Fi functionality, enabling users to remotely monitor surrounding conditions and manually manage the purification process as needed. The resulting system is scalable, energy-efficient, and optimized for interior settings, providing a practical solution for contemporary air quality control through AI-driven automation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
AI	Artificial Intelligence
AQI	Air Quality Index
CO_2	Carbon Dioxide
DHT11	Digital Humidity and Temperature sensor model 11
ESP32	Espressif Systems microcontroller 32-bit
ML	Machine Learning
PPM	Parts Per Million
PM2.5 / PM10	Particulate Matter 2.5 / 10 micrometers
UART	Universal Asynchronous Receiver Transmitter
Wi-Fi	Wireless Fidelity

CHAPTER 1

INTRODUCTION

Air pollution in cities has become a major public health problem because it causes more people to get lung diseases, heart problems, and other long-term illnesses. Indoors, where pollutants can build up to dangerous amounts, the problem is made worse. There are many air cleaning systems on the market, but most of them work continuously without checking the quality of the air in real time. This wastes energy and makes the cleaning process less effective.

The proposed system introduces a smart indoor air purification approach that combines Artificial Intelligence (AI) and Internet of Things (IoT) technologies to improve air quality. Data from several sensors, including the MQ-135 gas sensor for detecting harmful gases and the DHT11 sensor for measuring temperature and humidity, is gathered by the system using an ESP32 microcontroller. When pollution levels increase, the air is cleaned using a cabin air filter and DC fan to circulate air through a cabin air filter, effectively reducing airborne pollutants while also helping to regulate the temperature.

The method is further improved by creating a comprehensive dataset by combining real-time air quality observations with Open-Meteo meteorological data. A decision tree method that can forecast future trends in air quality was trained using this dataset. An ESP32 microcontroller running the trained model dynamically compares current sensor readings with projected values to modify the filter unit's operation for best results. To effectively eliminate airborne pollutants, a fan is connected to a cabin air filter, which is commonly utilized in vehicle applications. Furthermore, customers can remotely monitor and manage the purifying process thanks to the ESP32's Wi-Fi module, which permits real- time data transmission to a mobile interface.

The finished system offers an intelligent and automated method of managing air quality with AI integration, and it is scalable, energy-efficient, and suitable for contemporary interior environments.

1.1 MOTIVATION

The majority of individuals in today's urban lifestyle are indoors, whether at home, at work, or in enclosed public areas. Sadly, indoor air quality isn't always reliable. Allergies, respiratory issues, and exhaustion can be brought on by dust, toxic gasses, and inadequate ventilation. Conventional air purifiers waste energy and provide little flexibility because they frequently operate constantly, regardless of the actual quality of the air. This led to the creation of a refined, sensor-based air filtration system that uses artificial intelligence to react automatically in addition to monitoring indoor air. The objective is to develop a more intelligent, energy-efficient solution that enhances comfort and health by preserving clean indoor air with little human intervention.

1.2 REQUIREMENT SPECIFICATIONS

1.2.1 Functionality

- Monitor indoor air quality in real time using environmental sensors.
- Detect harmful gases and changes in temperature or humidity.
- Predict pollution patterns and react automatically using machine learning.
- Activate a purification system only when necessary.
- Provide manual control and live data access via a mobile/web dashboard.
- Work effectively and adjust to various room conditions.

1.2.2 Technology used

The system employs the ESP32 microcontroller for processing, control, and wireless communication. It integrates the MQ-135 sensor for detecting harmful gases and the DHT11 sensor for monitoring temperature and humidity. Air is purified using a DC fan and an automotive-grade cabin air filter. A decision tree model, trained using Google Colab, enables smart control based on environmental data. Real-time weather input from the Open-Meteo API further improves prediction accuracy. Initial design and behavior were tested using Proteus simulation software before hardware deployment.

1.2.3 SDG Goals

This project supports the following United Nations Sustainable Development Goals (SDGs):

- Goal 3 – Good Health and Well-being: Improves respiratory health by maintaining clean indoor air.
- Goal 7 – Affordable and Clean Energy: Reduces power usage through smart control of the purification system.
- Goal 11 – Sustainable Cities and Communities: Encourages the use of eco-friendly technologies in urban settings.
- Goal 13 – Climate Action: Supports energy conservation and environmental awareness by promoting efficient indoor air management.

CHAPTER 2

LITERATURE SURVEY

2.1 A REVIEW OF GENERAL AND MODERN METHODS OF AIR PURIFICATION

Roy et al. provided a detailed review of both traditional and advanced methods for air purification, addressing the growing concern over indoor and outdoor air pollution and its adverse effects on human health. The author categorizes purification methods into mechanical, chemical, biological, and emerging smart technologies, analyzing their operating principles, benefits, and limitations. Mechanical filtration methods, such as HEPA filters, are highlighted for their wide adoption and effectiveness in removing particulate matter, while chemical-based purifiers such as activated carbon and photocatalytic oxidation are examined for their ability to neutralize harmful gases and volatile organic compounds (VOCs).

Biological air purifiers that utilize microorganisms or plants to cleanse the air are also discussed, offering sustainable but slower alternatives. A significant portion of the paper is dedicated to emerging techniques like electrostatic precipitators, ionizers, and photocatalytic systems, which show promise in achieving high purification efficiency with lower energy consumption. These modern technologies are evaluated based on their adaptability to different environments, maintenance requirements, and ability to integrate with smart systems.

The review emphasizes that no single purification method is universally optimal; instead, hybrid systems combining multiple approaches often deliver the best performance. The author advocates for context-specific solutions factoring in pollutant types, space constraints, and economic feasibility especially in urban and industrial environments where pollution levels are dynamic and complex.

In conclusion, the study serves as a valuable resource for understanding the landscape of air purification technologies, presenting comparative insights into their applications and effectiveness. The findings suggest that a strategic combination of mechanical, chemical, and emerging smart technologies can lead to significant improvements in indoor air quality, offering a robust foundation for future research and innovation in this critical domain.

2.2 STUDIES ON NEW AIR PURIFICATION AND AIR QUALITY CONTROL SYSTEM OF AIRLINER CABIN

Hu et al. proposed a study on the development of an advanced air purification and air quality control system specifically designed for airliner cabins. Given the unique environmental conditions and confined spaces within aircraft, maintaining high air quality is critical for passenger comfort and health. The authors begin by identifying key challenges in cabin air management, including the accumulation of CO_2, volatile organic compounds (VOCs), airborne microbes, and unpleasant odors, which traditional systems often fail to effectively eliminate.

To address these issues, the study proposes a new hybrid purification system integrating physical filtration, chemical absorption, and plasma-based sterilization technologies. Each component is optimized to target specific pollutants: HEPA filters for particulates, activated carbon layers for chemical gases, and plasma modules for microbial and odor control. The system is designed with modular flexibility and automation to adjust purification intensity based on real-time air quality data.

Through a series of simulations and experimental validations conducted in controlled cabin-like environments, the proposed system demonstrates significant improvements in removing both particulate and gaseous pollutants. The results show enhanced microbial sterilization, reduced CO_2 levels, and a more consistent distribution of clean air throughout the cabin. The energy efficiency of the system is also evaluated, with findings indicating that the new approach consumes less power than traditional ventilation-heavy designs, while providing superior purification performance.

The paper also explores the integration of sensors and intelligent control algorithms to dynamically manage airflow and purification settings in response to occupancy levels and detected contaminant concentrations. This not only optimizes system efficiency but also extends component life and reduces maintenance frequency.

In conclusion, the study provides a forward-looking solution for improving cabin air quality through an intelligent, multi-layered purification system. The research offers valuable insights for future commercial aircraft design, emphasizing the importance of combining mechanical, chemical, and advanced sterilization technologies to ensure a healthier in-flight environment for passengers and crew.

2.3 FACTORS AFFECTING REAL-WORLD APPLICATIONS OF HEPA PURIFIERS IN IMPROVING INDOOR AIR QUALITY

Lowther et al. investigated the effectiveness of HEPA air purifiers in real-world indoor environments and examined the multiple factors that influence their performance beyond laboratory conditions. Recognizing that HEPA purifiers are widely recommended for mitigating indoor air pollution including allergens, particulate matter (PM), and airborne pathogens, the study aims to bridge the gap between controlled performance evaluations and practical deployment outcomes in homes, offices, and classrooms.

The authors analyze a wide range of variables that can impact the efficacy of HEPA filters, including room size, air exchange rates, purifier placement, occupant behavior, and indoor activity patterns. Experimental data were gathered from field trials across multiple residential and commercial settings. The results reveal that purifier performance can vary significantly due to suboptimal usage, such as incorrect sizing relative to room volume, poor maintenance (e.g., clogged filters), or inadequate run times.

A key finding of the study is the importance of strategic placement and continuous operation in achieving consistent air quality improvements. Devices placed near pollutant sources or in areas with better air circulation demonstrated higher particle removal efficiency. Furthermore, the study emphasizes the role of behavioral factors— such as opening windows, use of cleaning products, or cooking habits that can either complement or hinder purifier effectiveness.

The paper also discusses the energy consumption of HEPA purifiers and the trade-offs between operational costs and air quality gains. By modeling various use scenarios, the authors present practical recommendations for maximizing purifier efficiency while minimizing energy expenditure. They attest for integrating HEPA units with smart sensors and automated control systems that adjust fan speed and operation based on real-time indoor air quality metrics.

In conclusion, the research provides valuable insights into the real-world functionality of HEPA air purifiers, highlighting the complexity of indoor air dynamics and the necessity for user education and system optimization.

2.4 INDOOR AIR QUALITY IN A DOMESTIC ENVIRONMENT: COMBINED CONTRIBUTION OF INDOOR AND OUTDOOR PM SOURCES.

Luca et al. presented a detailed analysis of indoor air quality in residential settings, with a focus on the combined influence of both indoor-generated and outdoor-infiltrated particulate matter (PM). Recognizing that people spend the majority of their time indoors, the study aims to quantify and characterize the sources of PM10 and PM2.5 in domestic environments to better inform strategies for air quality improvement. The authors conduct real-time monitoring of particle concentrations in multiple households over extended periods, accounting for various activities, ventilation patterns, and ambient conditions.

A major contribution of the study is its methodology for distinguishing between indoor and outdoor PM sources using source apportionment models and time-resolved data. Common indoor contributors identified include cooking, smoking, cleaning activities, and candle burning, all of which generate significant short-term spikes in PM levels. Meanwhile, outdoor sources such as traffic emissions and urban dust are found to penetrate indoor environments depending on window use, building insulation quality, and HVAC system characteristics.

The study highlights that the indoor concentration of PM is not solely a function of external air quality, but also heavily influenced by occupant behavior and household practices. For example, homes with frequent cooking events or poor ventilation showed elevated PM levels regardless of outdoor pollution conditions. Conversely, well-sealed and mechanically ventilated homes exhibited better protection from outdoor infiltration.

Energy efficiency measures, such as airtight window designs, are shown to have a dual impact reducing pollutant entry but also trapping indoor emissions if not coupled with adequate ventilation. The authors stress the importance of balancing insulation with controlled air exchange to maintain healthy indoor environments.

In conclusion, the paper provides a comprehensive understanding of the dual origin of indoor particulate pollution, reinforcing the need for integrated solutions that address both indoor emission control and outdoor infiltration barriers. The findings support the development of smarter air management strategies in homes.

2.5. DESIGN AND DEVELOPMENT OF IOT BASED INDOOR AIR QUALITY MONITORING AND PURIFICATION USING HEPA FILTER

Waghmare et al. presented a practical approach to improving indoor air quality through the design and development of an IoT-based monitoring and purification system using a HEPA filter. With rising concerns over the health impacts of prolonged exposure to indoor pollutants, this study focuses on real-time air quality assessment and automated purification in domestic spaces.

The proposed system integrates various sensors to measure key indoor air quality parameters such as PM2.5, PM10, temperature, and humidity. These values are processed and transmitted via an IoT platform, enabling continuous monitoring and remote access to air quality data. A central feature of the system is its ability to automatically activate a HEPA filter-based purification mechanism when pollutant levels exceed safe thresholds, thereby maintaining healthier indoor conditions with minimal user intervention.

The authors emphasize the cost-effectiveness, scalability, and ease of deployment of their system, making it suitable for widespread use in urban households. Their findings demonstrate the effectiveness of combining sensor technology, wireless communication, and efficient filtration in addressing common indoor air pollutants generated by activities like cooking, smoking, and poor ventilation.

In conclusion, this work contributes to the growing field of smart environmental control systems by offering a user-friendly and responsive solution to indoor air pollution. It highlights the role of technology in fostering proactive air quality management, underscoring the potential for integrated IoT-based solutions to support healthier living environments.

2.6 SOLAR OUTDOOR AIR PURIFIER WITH AIR QUALITY MONITORING USING IOT

Nagendra et al. proposed an innovative solution to outdoor air pollution through the development of a solar-powered air purification system integrated with IoT- based monitoring. Their work aims to address the growing need for sustainable and autonomous technologies that can improve ambient air quality in urban environments.

The system is designed to operate independently using solar energy, making it suitable for deployment in public spaces, roadside areas, and pollution hotspots. Equipped with air quality sensors, the device continuously monitors pollutants such as PM2.5, PM10, and harmful gases, relaying real-time data to a cloud platform for visualization and analysis. When pollution levels rise above acceptable limits, the purifier—powered by solar panels and fitted with filtration components—is activated to clean the surrounding air.

One of the key contributions of this study is its emphasis on energy-efficient, eco-friendly operation without reliance on conventional power sources. The authors demonstrate that the system not only reduces pollutant concentration in outdoor air but also enables data-driven environmental management through IoT connectivity.

In conclusion, the paper presents a sustainable and scalable approach to outdoor air purification, highlighting the intersection of renewable energy, IoT, and environmental health. The work supports broader efforts to mitigate air pollution through smart infrastructure solutions tailored for modern urban challenges.

CHAPTER 3
SOFTWARE DESCRIPTION

3.1 PROTEUS DESIGN SUITE

Proteus Design Suite is a complete electronic design automation (EDA) tool often used for circuit simulation, PCB design, and embedded system development. It offers a flexible platform for creating and testing electronic circuits in a virtual environment, therefore reducing the need for immediate hardware prototyping.

Proteus Virtual System Modelling (VSM) capability is one of its most notable features, as it enables users to simulate the electrical behavior of a circuit and the actual firmware code that is running on supported microcontrollers[3]. This allows for the real-time debugging and validation of microcontroller-based systems, including those that utilize commonly used families such as Arduino, PIC, AVR, and ARM. The development process is significantly streamlined by the ability of users to write and simulate embedded code in conjunction with circuit design.

The software includes a vast library of components, such as microcontrollers, sensors, actuators, displays, and other peripheral devices. Its drag-and-drop schematic editor makes circuit construction intuitive, while its simulation engine provides accurate representation of analog and digital behavior. In addition to simulation, Proteus also offers powerful tools for PCB layout design, including features like auto-routing, design rule checks, and 3D visualization of the final board.

Overall, the Proteus Design Suite is a versatile and durable solution for the development of electronics. It combines circuit simulation, firmware debugging, and PCB design into a single platform, making it a valuable tool for both teaching and professional engineering workflows.

3.2 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is an open-source platform that is utilized to write, compile, and upload code to microcontroller boards, including the Arduino Uno and ESP32. It is essential in embedded systems, particularly when it comes to interacting with sensors and actuators[4]. The IDE enables the efficient management of real-time data collection from sensors such as the DHT11 (for temperature and humidity), PM2.5 sensors (for particulate matter), and MQ135 (for gas detection). It allows the setup of output devices, including fans, filters, and sprayers, and threshold-based decision-making. The code is designed to guarantee accurate logic execution, reliable sensor readings, and effective peripheral communication. The programming process is simplified by the use of custom libraries that are customized for each sensor. Additionally, the IDE's serial monitor facilitates real-time debugging and performance tracking. The Arduino IDE guarantees that the embedded system remains in sync with AI-driven decisions, thereby guaranteeing that air quality responses are accurate and timely.

3.3 GOOGLE COLLABORATORY

Google Colaboratory, commonly referred to as Google Colab, is a free, cloud-based platform developed by Google Research. Users can write and run Python code in a web- based Jupyter Notebook environment with Google Colab (Collaboratory), a cloud- based platform created by Google. Because it provides free access to computational resources like CPUs, GPUs, and TPUs, it is particularly well-liked for data science and machine learning tasks.In this project, a machine learning model for intelligent decision-making based on air quality was developed and trained using Google Colab.

Colab was an effective tool for iterative model development because it allowed for real-time collaboration, version control via Google Drive, and access to well-known Python libraries like scikit-learn, pandas, and matplotlib. Additionally, it offered a practical setting for testing various model configurations, debugging, and exporting results for deployment.

Google Colab was the perfect platform for creating the AI-based purification logic that adjusts to changing air quality conditions because of its simplicity, adaptability, and processing power.

CHAPTER 4

HARDWARE DESCRIPTION

4.1.DFROBOT FIREBEETLE 2 ESP32-S3

The DFRobot FireBeetle 2 ESP32-S3 is a compact and energy-efficient development board based on the ESP32-S3 microcontroller. Designed for modern embedded applications, it features dual-core processing, built-in Wi-Fi and Bluetooth 5.0 connectivity, and advanced AI support. Its architecture provides improved performance over conventional microcontrollers, making it suitable for tasks requiring fast data processing and flexible peripheral integration.

The board in figure 4.1 supports a wide variety of digital and analog interfaces, making it ideal for real-time data collection from multiple sensors and for controlling various output devices. Its ultra-low-power design, combined with integrated power management, makes it especially well-suited for battery-powered Internet of Things (IoT) applications[8]. In addition to strong wireless communication capabilities, the board is equipped to handle edge computing tasks, offering local processing to reduce dependence on cloud-based systems.

With its functionality and adaptability, the board serves as a powerful platform for building intelligent, low-power embedded solutions.

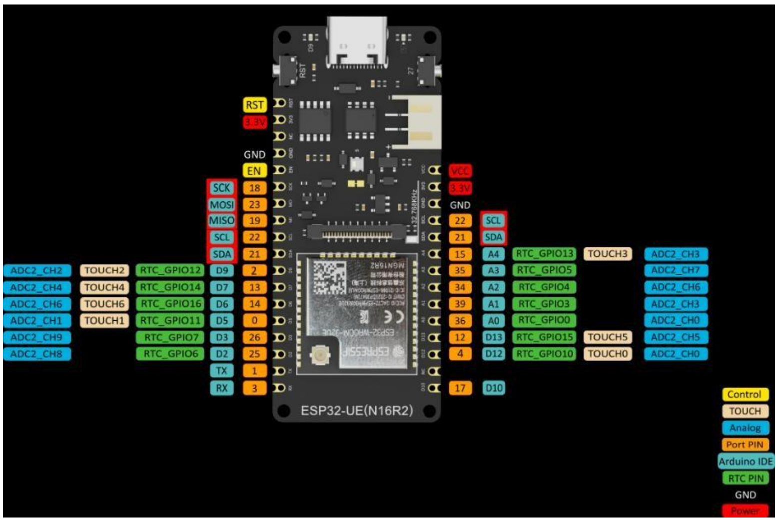

Figure 4.1 DFRobot Firebeetle 2 ESP32-S3

4.2. ARDUINO UNO

The Arduino Uno is a widely used open-source development board built around the ATmega328P microcontroller. It is specifically designed to simplify the process of building and programming digital systems, making it a popular choice in educational settings, hobby electronics, and rapid prototyping. Known for its user- friendly design, the Arduino Uno provides an accessible entry point for beginners while offering sufficient flexibility for more advanced users.

The board in figure 4.2 features a range of input and output capabilities, including digital and analog pins, PWM outputs, and communication interfaces such as I2C, SPI, and UART. It is typically programmed using the Arduino Integrated Development Environment (IDE), which supports a simplified version of C/C++ and offers an extensive set of built-in libraries. This allows developers to focus more on functionality and less on low-level hardware details.

Due to its reliability and ease of integration with sensors, actuators, and external modules, the Arduino Uno has been at the heart of numerous applications, ranging from robotics and home automation to environmental monitoring and interactive art installations. Its open-source nature encourages a strong global community of developers and learners, fostering continuous innovation and knowledge sharing. Overall, the Arduino Uno remains a fundamental platform in the embedded systems and maker ecosystem.

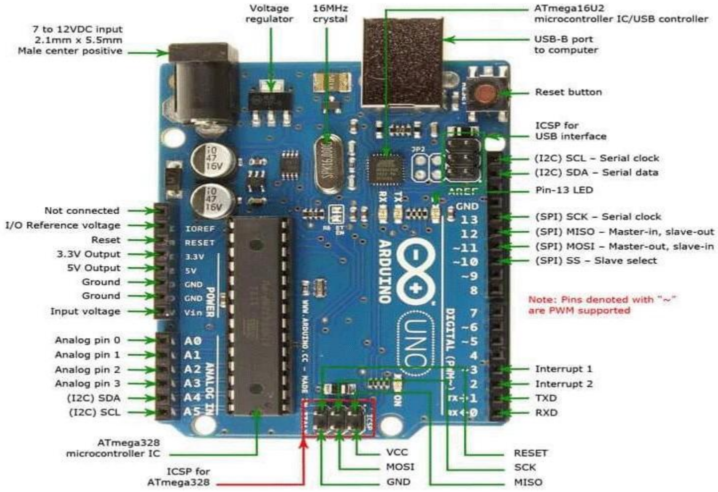

Figure 4.2 Arduino UNO

4.3. SPECIFICATIONS

FireBeetle 2 ESP32-S3

- Processor: Xtensa® dual-core 32-bit LX7 microprocessor
- Clock Speed: Up to 240 MHz
- Memory:
  - SRAM: 512 KB
  - ROM: 384 KB
  - Flash: 4 MB
  - RTC SRAM: 16 KB
- Connectivity:
  - Wi-Fi: IEEE 802.11 b/g/n (2.4 GHz, 20/40 MHz bandwidth, A-MPDU/A-MSDU support)
  - Bluetooth: Version 5, mesh support, speeds of 125 Kbps to 2 Mbps
- Interfaces:
  - Digital I/O: 26 pins
  - SPI: 2
  - UART: 3
  - I2C: 2
  - I2S: 2
- USB: USB 2.0 OTG full-speed interface
- ADC: 2 × 12-bit SAR ADC, 20 channels
- Other Features:
  - LED PWM: 8 channels
  - Infrared transceiver: 5 TX and 5 RX channels
  - DMA controller: 5 TX and 5 RX channels

- Power Input:
    - USB-C: 5V DC
    - PH2.0: 3.7V Li-ion battery
    - VCC pin: 5V DC
- Size: 25.4 mm × 60 mm
- Weight: 23.4 g
- Software Support: Arduino IDE, MicroPython, ESP-IDF

Arduino Uno R3:

- Microcontroller: ATmega328P
- Clock Speed: 16 MHz
- USB Connector: USB-B
- Memory:
    - SRAM: 2 KB
    - Flash: 32 KB
    - EEPROM: 1 KB
- I/O:
    - Digital I/O Pins: 14
    - Analog Input Pins: 6
    - PWM Pins: 6
    - Built-in LED: Pin 13
- Communication: UART, SPI, I2C
- Voltage:
    - I/O Voltage: 5V
    - Input Voltage (Recommended): 7–12V
    - DC Current per I/O Pin: 20 mA

- Power Connector: Barrel Plug
- Dimensions: 68.6 mm × 53.4 mm
- Weight: 25 g

4.4 AUXILIARY COMPONENTS

MQ-135 Gas Sensor

Figure 4.3: MQ 135 Gas Sensor

The MQ-135 is a widely used air quality sensor designed to detect a range of harmful gases, including ammonia (NH_3), benzene, carbon dioxide (CO_2), smoke, and alcohol vapors. It operates by measuring the change in resistance of its internal sensing layer in the presence of gases. The sensor in figure 4.3 provides an analog voltage output proportional to the gas concentration. With high sensitivity and fast response time, the MQ-135 is suitable for monitoring indoor air quality in applications like smart homes, air purifiers, and environmental sensing systems. It is typically interfaced with microcontrollers through an analog input pin.

DHT11 Temperature and Humidity Sensor

Figure 4.4: DHT Temperature Sensor

The DHT11 is a low-cost digital sensor used for measuring temperature and relative humidity. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and outputs a calibrated digital signal via a single data pin. The sensor in figure 4.4 offers a temperature range of 0°C to 50°C with ±2°C accuracy and a humidity range of 20% to 90% with ±5% accuracy. Its simple interface and stable performance make it ideal for indoor environmental monitoring and control systems where precise but non-industrial accuracy is sufficient.

DC Fan

Figure 4.5: DC Fan

The DC fan is used in the system as the primary actuator for air circulation and purification. It operates on a low-voltage DC supply (typically 5V or 12V) and is responsible for drawing air through the filter mechanism to remove particulates and pollutants. DC fan in figure 4.5 is controlled via a relay or directly by a microcontroller through PWM signals, the fan ensures that air movement occurs only when poor air quality is detected. Its compact size and energy efficiency make it suitable for use in embedded air purification systems.

Relay Module

Figure 4.6: Relay Module

A relay module serves as an electrically operated switch that allows the microcontroller to control high-power components, such as the fan, using low-power signals. It provides electrical isolation between the control circuit and the load using an electromagnetic mechanism. In this system, the relay in figure 4.6 ensures that the fan or other components can be turned on or off safely based on logic determined by air quality readings. Typically, it operates on 5V and supports normally open (NO) and normally closed (NC) contacts.

Car Cabin Air Filter

Figure 4.7: Cabin Air Filter

The car cabin air filter in figure 4.7 is repurposed in the proposed system as the main filtration unit. Originally designed for automotive use, it is capable of trapping dust, pollen, and other fine particles from incoming air. Its layered structure often made of activated carbon or microfiber materials provides effective physical filtration without requiring electronic control. When combined with a fan, the filter becomes an efficient low-cost solution for indoor air purification, making it ideal for compact systems that require passive air cleaning.

CHAPTER 5

METHODOLOGY

5.1. SYSTEM OVERVIEW

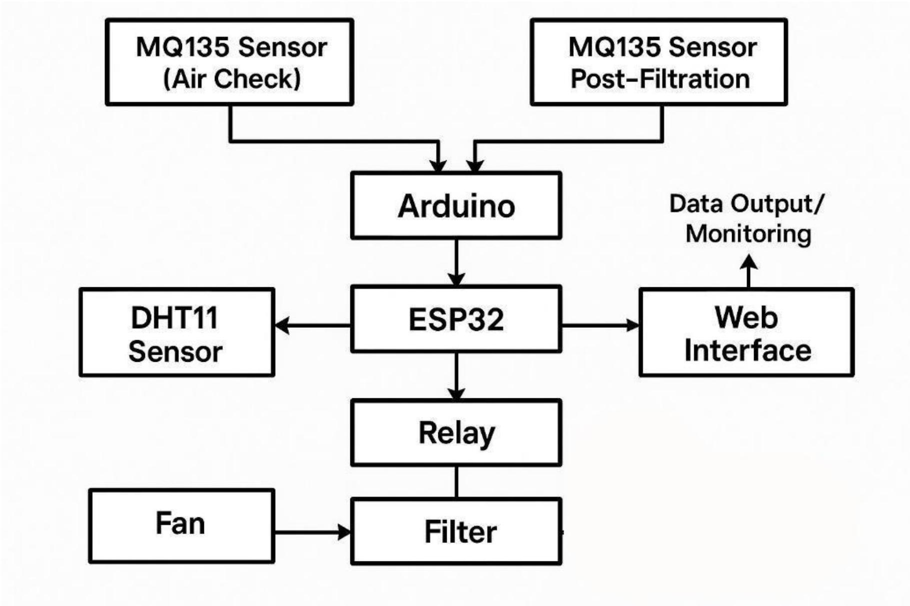

Figure 5.1: Block diagram

The above mentioned block diagram represents an air purification and monitoring system which is designed to ensure cleaner indoor air through real-time detection, filtration, and remote monitoring. The system in figure 5.1 initiates with two MQ135 gas sensors: one is positioned before the air filter to measure the purity of the incoming air, while the other is positioned after the filter to evaluate the efficiency of the purification process[6]. Various harmful gases, including ammonia, carbon dioxide, and smoke, are detected by these sensors.

Initially, the Arduino Uno processes the sensor data by collecting the analog readings and transmitting the processed information to the ESP32 microcontroller via serial communication. In addition to gas concentration data, the ESP32 also receives temperature and humidity readings from a connected DHT11 sensor, which allows for a thorough evaluation of the indoor environment.

The central controller, the ESP32, is responsible for management of both data interpretation and actuation. It transmits the environmental data that has been collected to a web interface via its integrated Wi-Fi module, thereby enabling users to remotely monitor the air quality in real time. On the basis of the sensor input, the ESP32 ascertains whether the air necessitates purification. A relay module is activated when pollutant levels exceed acceptable thresholds, which subsequently powers a DC fan and a cabin air filter.

The filter captures dust, smoke, and gaseous contaminants, while the fan introduces polluted air into the system. The air quality is subsequently reassessed by the post- filtration MQ135 sensor to confirm the results of the cleaning process. This closed-loop mechanism guarantees responsive air purification and enables users to remotely monitor and regulate the system. The system is an effective solution for indoor air quality management due to the integration of communication, control, and sensing.

5.2. DECISION TREE ARCHITECTURE

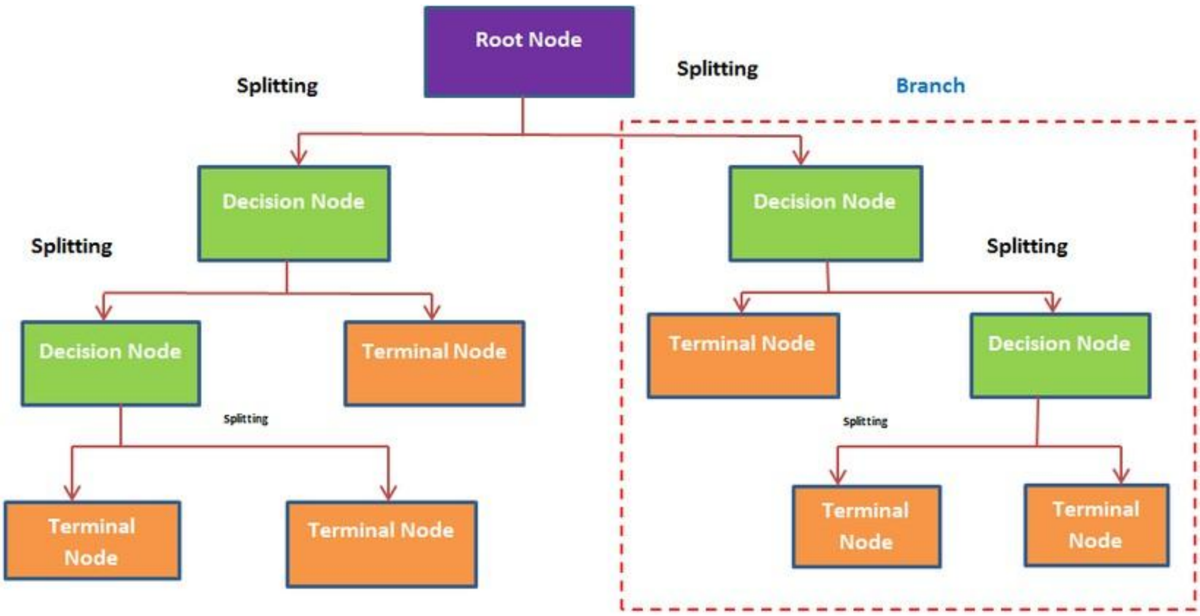

Figure 5.2: Decision Tree Algorithm Architecture

5.2.1 Overview

In the proposed air purification system, the decision-making logic is implemented using a Decision Tree algorithm in figure 5.2, trained on historical air quality and meteorological data. Each sensor reading such as gas concentration from the MQ-135 or temperature and humidity from the DHT11 is treated as an input feature that may influence the system's behavior.

The trained decision tree model generates a series of if-else rules, where each decision node represents a logical condition based on a sensor value (e.g., "Gas Level > Threshold"). These rules are then converted into a format suitable for embedded execution on the ESP32 microcontroller, typically as a .h header file containing nested if-else statements.

During real-time operation, the ESP32 begins evaluation at the root node of the decision tree. At each node, the system compares the incoming sensor data with a predefined threshold to determine the path—either left or right—toward the next node. This process continues until a leaf node is reached, which corresponds to a final decision output such as activating the fan or keeping the system idle.

The ESP32 executes this logic sequentially using conditional statements rather than machine learning libraries, ensuring low memory usage and fast inference. This method is ideal for embedded applications with constrained hardware resources.

For simple and shallow decision trees, this approach enables direct implementation using combinational logic (if-else structures). It eliminates the need for floating-point computations or runtime model interpretation, making it both energy-efficient and responsive key qualities for a real-time air monitoring and control system.

By using this lightweight, rule-based model, the purification system intelligently adapts to environmental changes without requiring continuous cloud access or high computation overhead, making it reliable for indoor deployment.

5.3. BASELINE ARCHITECTURE

A non-neural machine learning technique that recursively divides the input feature space to forecast class labels, the Decision Tree model used for the classification task unlike neural networks, the Decision Tree does not contain layers of neurons. Rather, it is a hierarchical system of nodes representing decision points depending on feature values.

Starting with a root node, the model gets all input features. The algorithm chooses the most informative feature and threshold value at every internal (non-leaf) node to divide the dataset using impurity criteria including the Gini Index or Information Gain. Until terminal nodes (leaf nodes) are reached, this recursive splitting carries on creating a tree structure. Every leaf node relates to a forecasted class label.

The mentioned tree-based architecture is intuitive and interpretable. The model can be visualized as a flowchart where each path from the root to a leaf represents a decision rule. The architecture adapts to the training data structure and can automatically handle both numerical and categorical inputs. The shape of the tree is defined by parameters like maximum depth and minimum samples per split. For instance, a tree with a maximum depth of 4 and binary splits can be abstractly represented as (Input Features → Decision Node1 → Decision Node2 → … → Leaf Node), with the depth controlling the number of decision layers.

The specified Decision Tree model is designed to make quick and interpretable predictions by evaluating a series of if-else conditions, making it highly efficient for structured data classification tasks.

5.3.1. Splitting Criterion: Mean Squared Error (MSE)

In regression-based decision tree models, Mean Squared Error (MSE) is used as the primary criterion for determining where and how to split the data. Unlike classification trees that use Gini Impurity or Entropy to evaluate purity, regression trees aim to reduce variance in the output values. MSE measures the average of the squares of the differences between actual and predicted values. A lower MSE indicates that the model is predicting values more accurately, making it a reliable metric for assessing the quality of each potential split.

When building the tree, the algorithm evaluates multiple split points across the features and calculates the MSE for the resulting child nodes. The predicted value in each node is the mean of the actual target values (e.g., pollutant level) within that node. The split that produces the lowest weighted average MSE across the child nodes is selected as the optimal decision point. This process is repeated recursively, allowing the tree to create increasingly accurate partitions of the data based on feature thresholds.

In the context of the given prototype in figure 5.3, MSE was essential for training a decision tree model that predicts air quality parameters like PM2.5 and PPM levels based on input from sensors such as MQ135, DHT11, and others [11]. By minimizing the prediction error at each split, the model ensures that system actions like activating fans or filters are based on precise and reliable air quality estimations. This leads to a smarter and more energy-efficient public air purification system.

$$MSE = 1/n \sum_{i=1}^{n} \quad (yi - y)^2 \qquad \textbf{(5.1)}$$

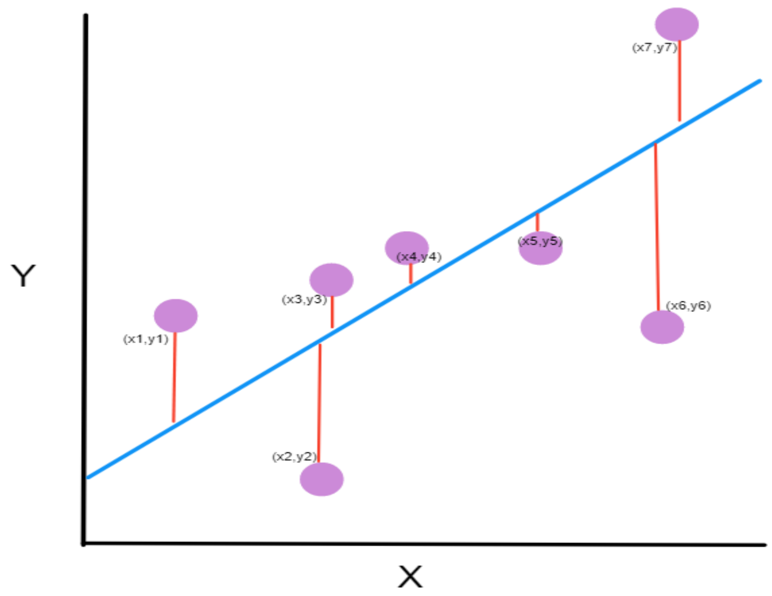

Figure 5.3 : Output graph

The key properties of the MSE are:

- Non-Negative: MSE is always greater than or equal to zero. It reaches its minimum value (zero) when the predicted values exactly match the true values.
- Differentiable: MSE is a smooth and continuous function, making it suitable for gradient-based optimization methods like gradient descent.
- Sensitive to Outliers: Since MSE squares the error, it penalizes larger errors more heavily, making it sensitive to outliers in the data.
- Convex Function: MSE is convex in nature with respect to the model parameters, ensuring that optimization algorithms like gradient descent will converge to a global minimum.

5.4. DATA GENERATION

A set of test data from a CSV file including AQI data for Chennai to evaluate the performance of the machine learning model. The dataset is arranged for quick processing and inference. Every entry is preprocessed and formatted to fit the model's input needs, including sensor readings such as air quality, temperature, and humidity. Adequate accuracy in the sensor values' normalization and encoding guarantees the model's precise predictions during evaluation. A text file holds the 784 binary values produced, one for every pixel.

5.5. ML INTEGRATION

The decision tree algorithm running the machine learning model is trained on historical sensor data from the MQ-135, DHT11 sensors, etc. during training and its performance is assessed on distinct data to prevent overfitting or underfitting. The training process minimizes prediction error by modifying the decision tree's structure, including splitting rules.

This approach ensures that the microcontroller can evaluate incoming sensor data, compare it with the model's thresholds, and predict air quality conditions instantly. Based on the prediction, the system can automatically decide whether to activate or adjust the air purification components. The lightweight structure of the model makes it ideal for embedded AI applications, offering fast, energy-efficient, and accurate inference directly at the edge.

5.6. ENGINEERING STANDARDS

1. ISO 16890 & ASHRAE 52.2

   - Standardized Testing for Air Filter Efficiency
   - These standards define test procedures and classifications for evaluating the efficiency of air filters in capturing particulate matter. ISO 16890 provides a global testing approach, while ASHRAE 52.2 focuses on U.S.-based filtration performance metrics using particle size removal efficiency.

2. IEEE Standard 1451

   - Smart Sensor Interoperability for Real-Time Air Quality Monitoring

   - This standard facilitates communication and interoperability between smart sensors and networks. It defines a common architecture and interface for smart transducers, supporting efficient and real-time environmental monitoring, including air quality.

3. ISO 50001

   - Energy Management System to Optimize AI-Driven Power Consumption

   - This international standard provides a framework for organizations to establish, implement, and improve energy management systems. It is crucial for AI-powered systems to ensure energy efficiency and reduce operational costs in real-time processing environments.

4. IEC 61508

   - Functional Safety for AI-Controlled Purification Systems

   - IEC 61508 outlines requirements for ensuring the functional safety of electrical and electronic systems, including those powered by AI. It is essential for designing reliable and fail-safe air purification systems where human health may be impacted.

5. ISO 14001

   - Environmental Management for Sustainability in Air Purification

   - This standard supports the development of effective environmental management systems (EMS), promoting sustainability. It is especially relevant for organizations developing purification solutions to reduce environmental footprint and ensure compliance with ecological regulations.

5.7. MULTIDISCIPLINARY ASPECT

- The core of the initiative is the IoT framework, which integrates sensors such the MQ-135 and DHT11 into the system to gather real-time data on humidity, temperature, and air quality. The embedded system (DFRobot Firebeetle 2 ESP32-S3) processes this data and uses it to operate the air purifier. Deployed on this IoT system, the decision tree model trained to forecast best actions automates air purification depending on real-time sensor data. The dashboard shows live values from the sensors, enabling real-time monitoring and user interaction to modify the behaviour of the purifier as required.
- Using historical sensor data, the decision tree model is trained to forecast based on new sensor inputs. Based on variables including air quality and environmental conditions, it decides when to turn on or modify the air purifier. Other optimization methods and stochastic gradient descent hone the model to minimize forecast inaccuracies. While the dashboard offers a user-friendly interface for monitoring, the model's predictions are combined with the IoT system, therefore affecting the real-time operation of the air purifier.
- The DFRobot Firebeetle 2 ESP32-S3 runs the machine learning model and controls the air purifier as the embedded system. Given its limitations, quick responsiveness and efficient processing are essential. The system constantly reads sensor data and modifies the air purifier to fit. Connected to the embedded system, the dashboard shows live sensor data so users can monitor performance in real-time and modify it depending on the continuous environmental changes.

CHAPTER 6

SIMULATIONS

6.1. PROTEUS SIMULATION

The Proteus simulation tool is used to virtually implement and test the embedded system design before deploying it to physical hardware. In the proposed simulation in figure 6.1, the embedded code developed for the Arduino UNO, along with components such as sensors and actuators, is imported into the Proteus environment. Proteus mimics the microcontroller and linked peripherals including the MQ-135 gas sensor, DHT11, DC fan, and power modules permitting real-time visualization and interaction. This guarantees the design works as intended before physical prototyping by means of comprehensive verification of circuit logic, control algorithms, and system behavior under several conditions.

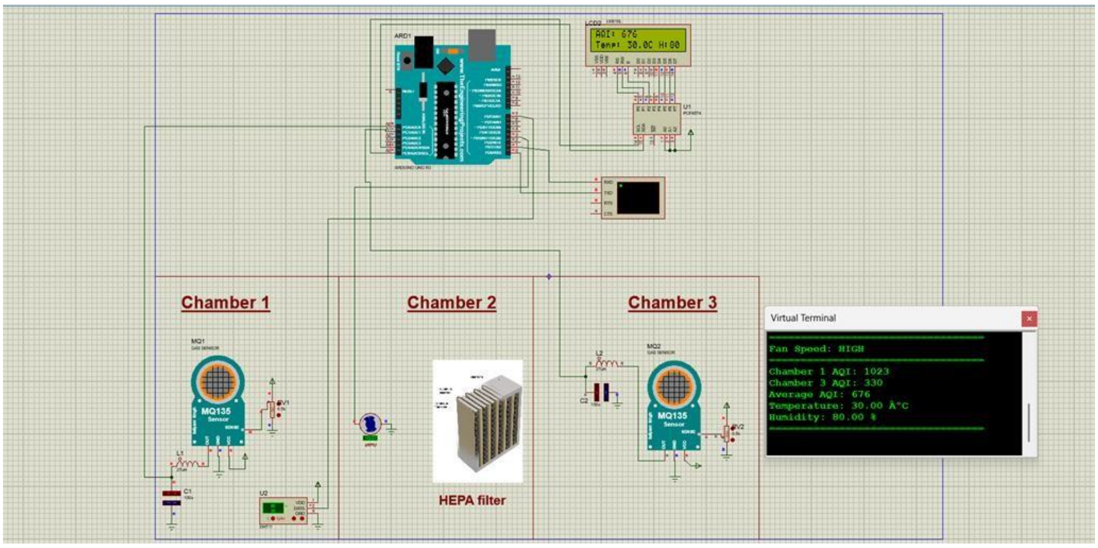

Figure 6.1: Software Simulation

Set up to examine and filter air, three chambers in this simulation show an Arduino microcontroller-based air quality monitoring and filtration system. While Chamber 2 has a HEPA filter for air purification, Chambers 1 and 3 are fitted with MQ135 air quality sensors to gauge parameters including ammonia, alcohol, benzene, smoke, and CO2 levels. Real-time air quality readings and environmental conditions (like temperature and humidity) before and after filtration are shown on an LCD and a virtual terminal by the Arduino, which gathers sensor data, processes it, and displays the findings. With the system offering ongoing monitoring and feedback, this configuration shows how air quality changes as it travels through the HEPA filter.

CHAPTER 7
RESULTS AND INFERENCES

7.1. INFERENCE

An AI-based air purification system was built using a DFRobot Firebeetle 2 ESP32-S3 microcontroller combined with Arduino and a decision tree machine learning model, as earlier mentioned. Environmental sensors linked to the microcontroller gathered real-time air quality data. On a labeled dataset reflecting different degrees of air pollution, Python trained the decision tree model. Multiple test runs in different air conditions were done to confirm the response and accuracy of the system after deploying the trained model onto the embedded system. By consistently classifying air quality levels correctly and triggering purification processes accordingly, the system showed the practical viability of including ML models into low-power IoT devices for real-world environmental monitoring applications.

Table 7.1: Data Set used for the ML model

S.NO	DATE	TEMP_MIN	TEMP_MAX	HIMIDITY_MEAN	PM25	PPM
1.	1/4/2025	28.7	71	40	40	1058.81
2.	2/4/2025	27.8	70	42.5	42.5	459.12
3.	3/4/2025	28.2	69	45	45	614.36
4.	4/4/2025	29	71	47.5	47.5	1027.88
5.	5/4/2025	29	71	50	50	562

Comprising five entries with environmental variables in table 7.1 gathered over successive days, the dataset every entry has the date, mean relative humidity (in %), PM2.5 concentration (in µg/m³), minimum and maximum temperatures (in °C). The data kinds are suitably formatted: humidity is shown as an integer, while temperatures and PM2.5 values are logged as floating-point numbers. There were no missing values noted; all measurements were within reasonable environmental ranges. Especially with decision tree models, additional preprocessing techniques such as normalization of numerical features e.g., temperature and PM2.5 might be used for efficient application in a machine learning pipeline. Extra feature engineering can improve the performance of the model by deriving temperature range or classifying PM2.5 levels into categories (like Good, Moderate, Poor).

For classification or environmental forecasting purposes, the dataset is tidy and well-organized.

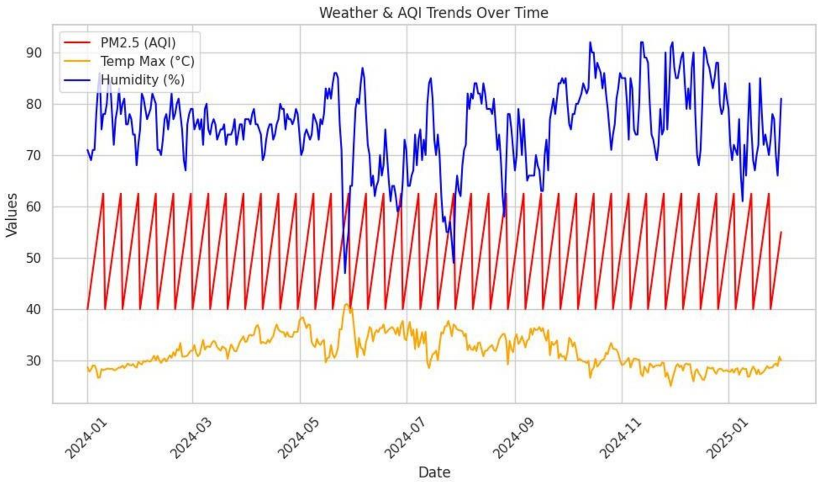

Figure 7.1: Weather & AQI Trends Over Time

The graphical representation shown in figure 7.1 illustrates the trends of key environmental factors PM2.5 (AQI), maximum temperature, and humidity over time. This visualization supports the understanding of how the trained machine learning model uses these features to classify air quality. By observing patterns and fluctuations in these variables, we can interpret how the model identifies important thresholds for decision-making. For instance, consistent spikes or drops in temperature or PM2.5 levels may act as critical indicators in the model's classification process. This kind of visual analysis enhances transparency, making it easier to validate the model's logic and ensuring its reliability in real-world air purification scenarios.

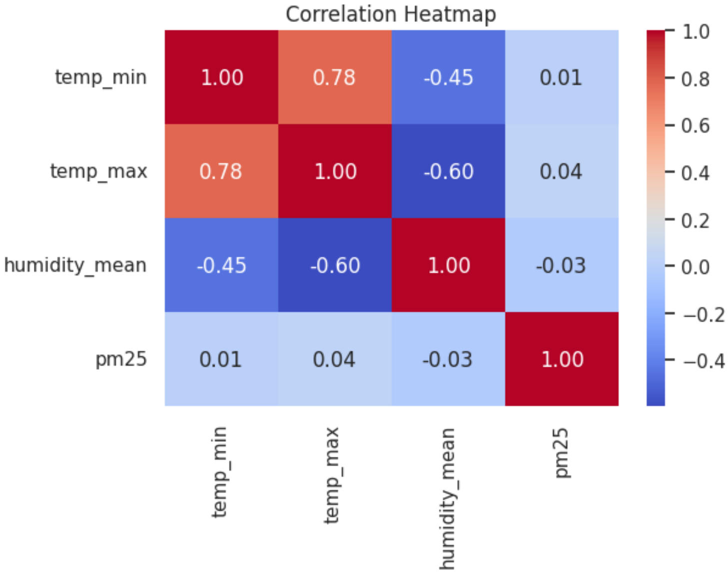

Figure 7.2: Correlation Heatmap

The correlation heatmap in figure 7.2 provides a visual representation of the relationships between key environmental variables minimum temperature, maximum temperature, mean humidity, and PM2.5 levels. Strong positive or negative correlations are highlighted by the intensity of the colors: red indicates a strong positive correlation, while blue shows a strong negative correlation. For example, minimum and maximum temperatures are highly positively correlated (0.78), while maximum temperature and humidity show a notable negative correlation (–0.60). On the other hand, PM2.5 appears to have very weak correlations with the other variables, suggesting that its fluctuations are not strongly influenced by temperature or humidity alone. This insight is useful in understanding which features significantly interact and which are more independent, aiding the machine learning model in identifying the most relevant predictors for accurate air quality classification.

7.2. HARDWARE SETUP

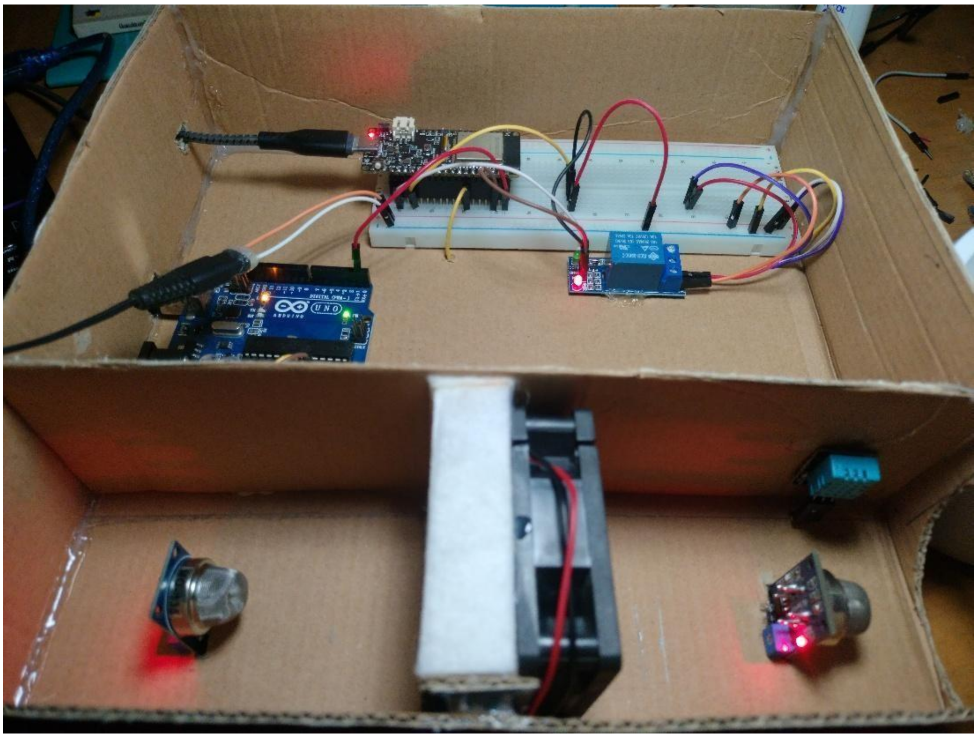

Figure 7.5: Hardware setup of the prototype

An AI-based air purification system's simplified prototype is represented by the hardware configuration in the figure 7.5. For demonstration and structural support, it is constructed inside a cardboard enclosure. As the primary processing unit, the DFRobot FireBeetle 2 ESP32-S3 board is at the heart of the system. It is in charge of classifying air quality levels in real time using the trained decision tree machine learning model. To manage analog sensor data and regulate signal distribution, an Arduino Uno board is also included. To make power supply and component wiring easier, both boards are connected to a single breadboard.

To measure environmental parameters, a number of sensors are positioned strategically. While MQ 135 gas sensors identify particulate matter or gaseous pollutants like ammonia or carbon monoxide, a DHT11 sensor keeps an eye on temperature and humidity levels. The ESP32-S3 receives the gathered sensor data and analyzes it. The system determines whether the air quality is within acceptable bounds based on the classification output from the decision tree model. A relay module is triggered to start the air purification process when pollutants surpass a predetermined threshold.

The setup's lower chamber displays the fan system that is part of the air purification mechanism. The relay activates this fan, which circulates air through a filter to eliminate particulate matter and enhance the general quality of the air in the prototype environment. Sensor responses and system activity are shown by LEDs on the boards and sensors. This hardware configuration is appropriate for smart environment applications since it effectively illustrates how machine learning and embedded systems can be combined to automate real-time air quality assessment and purification.

7.3. DASHBOARD

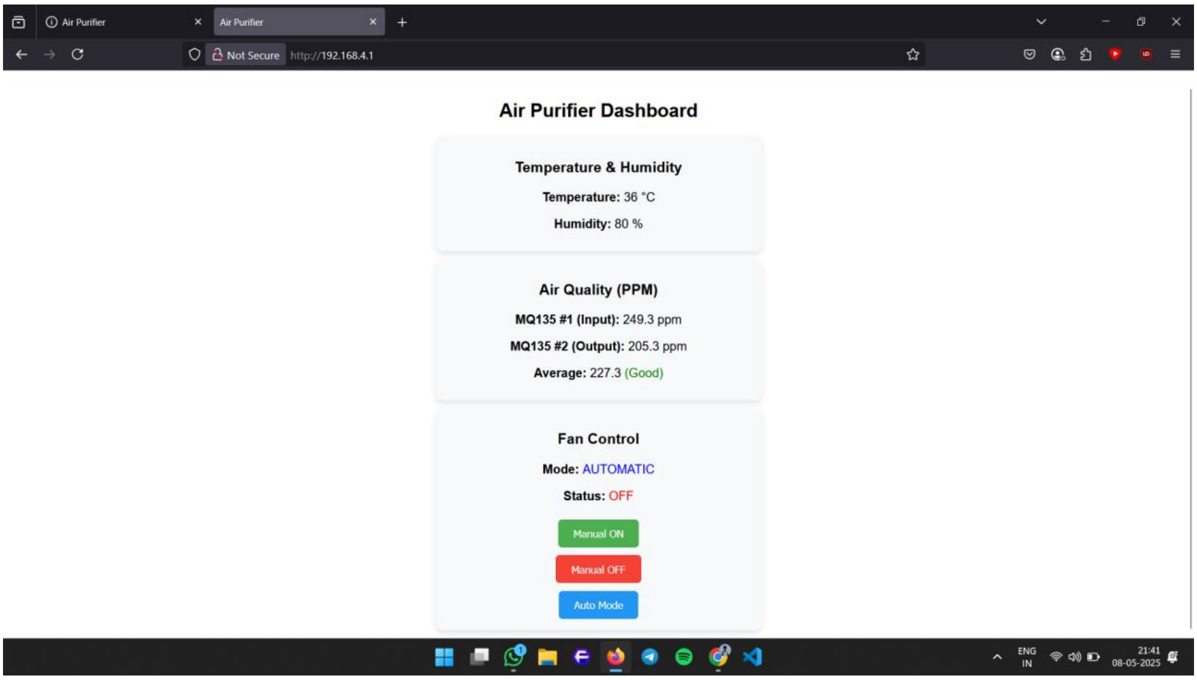

Figure 7.6: Dashboard for displaying the results & control specific components

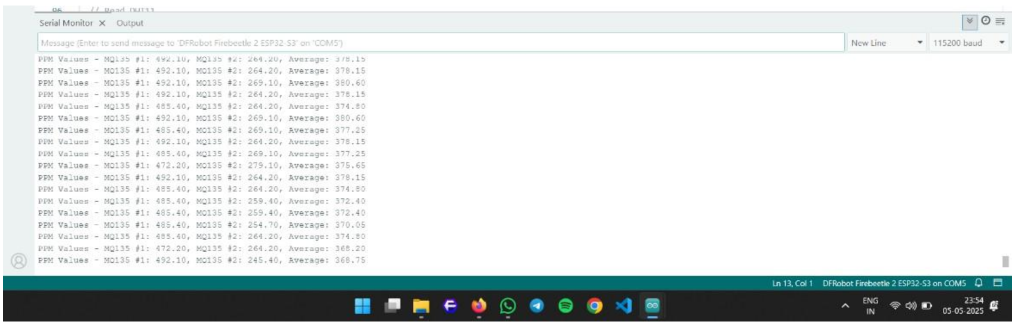

Figure 7.7 : Serial Monitor Output

The dashboard displayed in figure 7.6 was developed using the onboard Wi-Fi capabilities of the DFRobot FireBeetle 2 ESP32-S3 board, enabling real-time monitoring and control of the air purification system over a local network. The web interface shows crucial environmental parameters such as humidity, temperature, and air quality readings from two MQ135 gas sensors. These readings are displayed in Parts Per Million (PPM), along with their average value, providing users with clear insights into the current air quality levels which is given in figure 7.7. The system is capable of switching between different fan modes automatic and manual based on these sensor inputs and the output of the trained machine learning model.

Additionally, the dashboard lets users turn the fan on or off manually or activate Auto Mode, which employs the ESP32-S3's decision tree model to intelligently control fan operation. Users can interact with the air purifier system in a seamless manner thanks to this user interface's accessibility and simplicity. This project shows how embedded systems, IoT, and machine learning can be combined to create a smart, user-responsive air quality management system by integrating wireless communication and a web-based control panel.

CHAPTER 8
CONCLUSION AND FUTURE WORK

8.1.CONCLUSION

Using the DFRobot Firebeetle 2 ESP32-S3 microcontroller, Arduino environment, and a decision tree machine learning model, the application of an AI-based air purification system shows a hopeful way to real-time environmental monitoring and purification. Achieving efficient classification of air quality data with least computational overhead, the decision tree model offers understandable and lightweight logic appropriate for microcontroller-based applications. Particularly in limited embedded environments, the project successfully demonstrates how edge artificial intelligence can be used to enhance air purification processes.

8.2.FUTURE WORK

The proposed system can be enlarged in the future by investigating more sophisticated machine learning models like Random Forest or gradient boosting, which could provide better classification accuracy if supported by the hardware. More thorough environmental analysis can be achieved by adding more air quality sensors, including those for various gases or particles. Real-time user feedback systems such as mobile app alerts or visual cues can be created to improve usability. Especially for battery-powered applications, optimizing data sampling intervals and using low-power modes of the ESP32-S3 can help to increase energy efficiency. Moreover, allowing cloud connection would let remote monitoring, trend analysis, and long-term data storage, which could help more intelligent and scalable air purification systems.

8.3.REALISTIC CONSTRAINTS

- Though strong, the ESP32-S3 microcontroller's limited memory and processing power limits the size and complexity of the data handling logic and machine learning model.
- The efficacy of classification and response systems is greatly influenced by sensor accuracy and calibration; inexpensive sensors could add noise.

- Real-time data collecting or Wi-Fi-based operations could be limited under power restrictions, especially for portable or battery-operated systems.

- Without periodic recalibration or model retraining, environmental variability and sensor drift over time could lower the accuracy of predictions.

- Integration with external purification equipment might call for careful adjustment to guarantee safe and consistent operation under various air quality conditions.

8.4 APPLICATIONS

The developed AI-based air purification system is highly adaptable and can be implemented across a range of indoor environments where air quality monitoring and control are critical. Potential applications include:

- **Residential Homes**: To ensure safe breathing environments, especially for children, the elderly, and individuals with respiratory conditions.

- **Offices and Workspaces**: To promote healthier indoor air for employees, potentially improving concentration and productivity.

- **Schools and Educational Institutions**: To maintain optimal air quality in classrooms and reduce exposure to airborne pollutants among students.

- **Healthcare Facilities**: As a low-cost, energy-efficient supplement to existing ventilation systems in clinics or patient rooms.

8.5 LIMITATIONS

While the proposed system successfully combines AI and IoT for indoor air purification, there are still a number of issues that could compromise its functionality or acceptance in more general applications, such as Restricted Complexity of the Model. Although the decision tree model is lightweight and appropriate for embedded systems, it might not be as good at handling complex or highly non-linear patterns as more sophisticated models like neural networks or Random Forests. Sensor Accuracy and Sturdiness Despite being reasonably priced, the MQ-135 and DHT11 sensors have poor accuracy and could deteriorate over time. Additionally, they are susceptible to changes in humidity and temperature, which may compromise the accuracy of the data.

CHAPTER 9

REFERENCES

[1] Swamy, G. S. N. V. K. S. N. "Development of an indoor air purification system to improve ventilation and air quality." Heliyon 7.10 2021.

[2] Nasution, Tigor Hamonangan, et al. "Design of Indoor Air Quality Monitoring Systems." 2020 4rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM). IEEE, 2020.

[3] Shrimandilkar, P.P. Indoor air quality monitoring for human health. International Journal of Modern Engineering Research, 3(2), 891-897, 2013.

[4] Roy, Aditya. "A review of general and modern methods of air purification." Journal of Thermal Engineering 5.2: 22-28, 2019.

[5] Tao, Hu, et al. "Studies on new air purification and air quality control system of airliner cabin." Procedia Engineering 17: 343-353, 2011.

[6] Lowther, Scott D., et al. "Factors affecting real-world applications of HEPA purifiers in improving indoor air quality." Environmental Science: Advances 2.2: 235-246, 2023.

[7] Tofful, Luca, et al. "Indoor air quality in a domestic environment: Combined contribution of indoor and outdoor PM sources." *Building and Environment 202: 108050, 2021.

[8] M. Shinde, S. Kadam, and N. Waghmare, "Design and Development of IOT based Indoor Air Quality Monitoring and Purification using HEPA Filter," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 11, no. 8, pp. 102–104, Aug., 2024.

[9] Dr. Nagendra Kumar M, Darshan K Gowda, Suprith S, Vinay N, "Solar Outdoor Air Purifier with Air Quality Monitoring Using IoT," International Journal of Advanced Research in Computer and Communication Engineering, vol. 12, no. 5, pp. 1180-1183,. doi:10.17148/IJARCCE.2023.125200, 2023.

[10] Madhavi Shinde, Saurabh Kadam, Nahush Waghmare, "Design and Development of IOT based Indoor Air Quality Monitoring and Purification using HEPA Filter," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 11, no. 8, pp. 102-104, 2024.

[11] Z. Zhao, Y. Wang, L. Chen, "Integration of HEPA Filters in IoT-Based Air Purification Systems: Efficacy and Applications," IEEE Access, vol. 6, pp. 11013-11024, 2022.

[12] Efficacy of HEPA Air Cleaner on Improving Indoor Particulate Matter Concentration in Homes: A Randomized Crossover Study," International Journal of Environmental Research and Public Health, vol. 19, no. 18, Art. no. 11517, 2022.

[13] S. S. Dhingra, P. Singh, et al., "Smart Air Purifier with Air Quality Monitoring System," International Research Journal of Engineering and Technology (IRJET), vol. 9, no. 11, pp. 185–189, Nov. 2022.