2025 ASEE Northeast Section Conference, March 22, 2025, University of Bridgeport, Bridgeport, CT, USA.

AI-Driven Wildfire Detection with Integrated Air Quality and Machine Learning Vision Systems Powered by an Accelerator for Early Action

Mina Gaber Wahba Ibrahim*,
Department of Electrical Engineering
University of Bridgeport
Bridgeport, CT 06604, USA
*Corresponding author, Email: gabermina@gmail.com

Xingguo Xiong, Navarun Gupta, Ahmed El-Sayed
Department of Electrical and Computer Engineering,
University of Bridgeport,
Bridgeport, CT 06604, USA

***Abstract*—Wildfires are causing tremendous threats to human lives, infrastructure, the environment, and human health, entail innovative approaches for prompt detection and control. This research presents a state-of-the-art wildfire detection system that integrates an AI accelerator with air quality data and machine learning systems to identify potential fire danger. Utilizing real-time ecosystem data, including humidity, PM, VOC, RH/T, and NO, the system predicts the likelihood of wildfire ignitions. Upon a potential wildfire identification, the AI accelerator activates vision-based systems. Combining the power of AI accelerator and Google Colab, this approach achieves an efficient, and scalable solution for wildfire detection. By integrating air quality monitoring and AI-driven vision systems, this research provides a scalable, real-time wildfire detection solution.**

***Keywords*—air quality sensors; Arduino; Flask server; particulate matter; Hailo AI-Accelerator; Wildfire risk.**

I. Introduction

Wildfire poses a tremendous and escalating threat to human lives, infrastructure, ecological balance, and public health around the world. For instance, In 2025, wildfires in Los Angeles devastated neighborhoods, with 7,900 buildings damaged in the Eaton Fire [1]. The continuous increase and intensity of previously mentioned events, provided and derived by climate change and altered land management practices, open the door for the development of innovative and robust study for early detection systems. In 2022 4,304,379 Acres burned in California while 11,116 structures destroyed, as well as 1,050,012 acres burned with 2,148 structures destroyed in 2024 [3] and still occurs in 2025. Traditional wildfire detection methods, for example, satellite imagery and ground-based visual monitoring, and even advanced real time systems such as NASA FIRMS [4] usually suffer from limitations in timeliness and spatial resolution, especially in the critical early stages of fire ignition and smoke turbulence and convection current. This research investigates the urgent need for enhanced early wildfire detection systems using presenting a multistage system that integrates advanced air quality monitoring with the AI-accelerated vision systems that is wirelessly connected to the sensor microprocessor unit.

The system collects real-time environmental data, including particulate matter (PM1.0, PM2.5, PM4.0, PM10.0), volatile organic compounds (VOCs), nitrogen oxides (NOx), temperature, and humidity, to assess wildfire risk [5]. For example, the author in [6] emphasizes the importance of understanding PM2.5 emissions from biomass burning and particularly in the context of prescribed burns. These factors measured with high accurate reading using the SEN55 air quality sensor, provide comprehensive data of the immediate environment, that enable the identification and detection of potential fire hazards events before they escalate. The leveraging of the computational power of the Arduino Uno R4 WiFi [7] for data acquisition and the Raspberry Pi 5 [8] for data processing, the system recognizes a robust foundation for real time wildfire monitoring for early detection that triggers an AI-Driven system.

This paper essentially focuses on the initial development and evaluation of the air quality monitoring elements identified, including the hardware and software setup, data acquisition and transmission, as well as experimental validation. The results achieved demonstrate the system's sensitivity and reliability to the occurrence of fire and smoke events, showcasing its potential as a significantly important tool for early control wildfire detection. Future work will expand and develop upon this foundation by integrating an AI accelerator 'Hailo' [9], which a promising result has been achieved for advanced data analysis and a vision based system for fire verification, culminating in a comprehensive, scalable, robust, and efficient wildfire detection solution for humanity.

II. Methodology

A. System architecture

This research uses multistage process wildfire detection system designed for early action. This section will focus on the hardware setup, data collection process, experimental design and challenges encountered during the implementation. The system comprises four main components: (1) an air quality sensor (SEN55), (2) a data transmission unit (Arduino Uno R4 WiFi), (3) a central processing unit (Raspberry Pi 5), and (4) a vision-based AI accelerator.

Fig. 1. Shows the SEN55 sensor by Sensirion that is used during the experiments as the air quality sensor that is capable of sensing PM1.0, PM2.5, PM4.0, PM10.0, VOC, NOx, and ambient temperature.

Equipped with Wi-Fi capabilities for real-time data transmission to a remote server. A 1kΩ pull-up resistor was added to the I2C communication lines to ensure stable data transfer between the SEN55 sensor and the Arduino. The second component is the data transmission which is done by transmitting the sensor data wirelessly to a central processing unit using the ESP32 IC of the WiFi capabilities of the Arduino and the HTTP POST protocol. Third part which is the Central processing unit Raspberry Pi 5 which Acts as the central server for receiving and processing sensor data. Hosts a Flask-based web server to handle HTTP POST requests from the Arduino. Stores and analyzes data for further processing and visualization. AI accelerator (Hailo) will eventually integrate to enable real-time machine learning inference for fire prediction and will be used to process data from both the air quality sensor and the vision system. The fouth component is the vision system and that will be done using a camera system that could be connected to a drone or other systems. A camera module will be integrated to provide visual confirmation of fire events detected by the air quality sensor.

The fourth component is currently under development and will be covered in future work.

B. Hardware Implementation

Sensor Unit the SEN55 is a high-accuracy multi-parameter air quality sensor capable of measuring particulate Matter PM1.0, PM2.5, PM4.0, and PM10 'µg/m³'. Humidity: Relative Humidity '%'. Temperature: Ambient Temperature '°C'. Volatile Organic Compounds 'VOCs': Index value. Nitrogen Oxides 'NOx' Index value. The Functional block diagram illustrating in Fig. 2.

The SEN55 sensor communicates with the Arduino Uno R4 WiFi via the I2C communication protocol. The core of the sensor unit is the Arduino Uno R4 WiFi microcontroller designed around the 32-bit microcontroller RA4M1 from Renesas while also featuring an ESP32 module for Wi-Fi® and Bluetooth® connectivity showing in Fig. 3. This board was selected due to its integrated WiFi connectivity and compatibility with the SEN55 sensor.

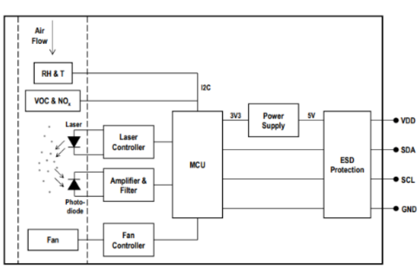

Fig. 2: Showing the Functional block diagram of the SEN55. Diagram by Sensirion that is clarify the environmental quality sensor components.

Configuration of the Pullup Resistor required for the air quality system needed. The SEN55 sensor operates at 5Vdc, I2C communication on the Arduino Uno R4 WiFi is default to 3.3V. To modify proper communication connection, external 1kΩ pullup resistors were added to the SDA and SCL trace of the I2C bus showing in Fig. 2 , connecting them to the 5V power supply. This solves the voltage mismatch and allows data to transfer between the sensor and the microcontroller. previous attempts using the ESP-Wroom32, which operates at 3.3Vdc, were unsuccessful due to this voltage incompatibility.

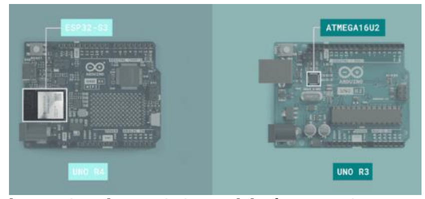

Fig. 3: showing arduino R3 and R4 ESP32 module for Wi-Fi®. Image from Arduino Webpage [10].

Different WiFi boards were tested to choose the most suitable board for the study. Prior to choosing the Arduino Uno R4 WiFi, other WiFi enabled boards were investigated. The ESP-Wroom32 was considered but was incompatible due to the 5V requirement of the SEN55. The xPico WiFi module was also explored, however, its complex configuration and integration with the sensor demonstrate challenging within the project timeframe.

Central Processing Unit chosen for the study is a Raspberry Pi 5, featuring the Broadcom BCM2712 quad-core Arm Cortex A76 processor @ 2.4GHz, making it up to three times faster than any previous generation [11]. Its computational power and network connectivity make it a great choice for receiving and processing the sensor data, while managing the Flask server, and finally integrating with the AI accelerator and vision system.

C. Data Transmission and Processing

Arduino Code main in this section provides a simplified version of the Arduino code, highlights the key parts of the sensor reading, data formatting, and HTTP POST [12] request with explanation for the libraries used ‘SensirionI2CSen5x, WiFiS3’.

The Arduino code written in C++ is responsible for reading data from the SEN55 sensor, formatting it, and sending it to the Raspberry Pi board using the WiFi. Essential factors of the code are Initialization of the SEN55 sensor using the SensirionI2CSen5x library. Cyclic reading of sensor values using the readMeasuredValues() function. Data formatting into a string containing key-value pairs ‘particulate matter (PM1.0, PM2.5, PM4.0, PM10.0), volatile organic compounds (VOC), nitrogen oxides (NOx), ambient temperature, and relative humidity’. Establishing a connection to the Raspberry Pi server by the WiFiS3 library. Sending the formatted data collection via an HTTP POST request to the predefined server IP address and port.

```cpp
#include <Arduino.h>
#include <SensirionI2CSen5x.h>
#include <Wire.h>
#include <WiFiS3.h>
// Wi-Fi setting
const char* ssid = "**";
const char* password = "**";
// Raspberry Pi server
const char* server = "192.168.0.20";
const int port = 8080;
WiFiClient client;
SensirionI2CSen5x sen5x;

float massConcentrationPm1p0, massConcentrationPm2p5,
      massConcentrationPm4p0, massConcentrationPm10p0;

float ambientHumidity, ambientTemperature, vocIndex, noxIndex;
```

Fig. 4. Shows the code snip illustrates the libraries and structure of the code written in C++ and uploaded to the Arduino microprocessor with the WiFi settings and the raspberry pi server. The variable used to read the data from the SEN55 sensor is in float data type.

Raspberry Pi Server used is a Flask-based web server developed to receive and process all the sensor data. The server logs the data and implies basic analysis, such as detecting abnormal spikes in particulate matter concentration.

Error received while installing Flask server due to Python's externally managed environments feature, which is designed to stop conflicts between system managed and user installed Python packages. Common errors in newer versions of Python '3.11 and later' and on Raspberry Pi OS System. Virtual environment successfully fixed the externally managed library error. A virtual environment gives access to install Python packages in an isolated environment without causing any effect to the system wide Python installation.

```python
from flask import Flask, request

app = Flask(__name__)

@app.route('/data', methods=['POST'])
def data():...
if __name__ == '__main__':
    # Run the Flask app
    app.run(host='192.168.0.20', port=8080)
```

Fig. 5 shows code snip for Flask Server Script showing the IP address and the port written to match the Arduino code.

AI Accelerator will be continued and presented as upcoming work. The Hailo AI accelerator is used to train machine learning models for real time fire prediction and detection. This component is in the implementation phase at the moment and will be integrated in future work.

Data transmission using the 'HTTP POST' protocol which provides efficient and reliable integration. Previously explored the performance with MQTT and TCP/IP. The TCP/IP was giving updates from sensor data with high cycle time. The MQTT is reputable for its performance for the real time sensor data transfer, but for this research was Inadequate and there was unnecessary for sending the data to the server and encrypted it as Flask based web server gives a simple with high speed outperformance with simple integration between the Arduino Wi-Fi and the raspberry pi 5.

D. *Experimental Setup*

For ethical consideration all experiments were conducted in a controlled environment to ensure safety and minimize the environmental risk by wearing the proper PPE and preparing the fire stop equipment in case needed. And all the experiments were supervised to prevent accidents.

The outdoor experiments were conducted on a rainy day. Wildfire experiments were conducted under cold and rainy conditions. The aim of these experiments was to collect sensor data under controlled conditions with simulated wildfire scenarios. Fire simulated with Small and controlled fires were created using kitchen lighter. Different tree types were used to simulate various wildfire conditions. The experiments indoor were conducted under room temperature.

A small pile of dry maple leaves 'approximately 20 grams were placed on a non-combustible surface. The leaves were ignited using a standard butane lighter Fig. 6. The flame was allowed to propagate naturally through the pile to the point of the leaves were fully consumed.

Fig. 6. Shows the dry maple leaves ignited and the embers sloping toward the wind direction and detected by the SEN55 that was placed in the left side of the setup. The dry leaves were placed on a non-combustible surface.

Small pieces of red cedar kindling approximately 30 grams were arranged in a loose pile shown in Fig. 7. The red ceder burned while shut off quickly and the igniter flame applied for the duration of the experiment due to the flame not being stay on.

Fig. 7. Shows the red cedar tree branch was unsuccessfully ignited. A small bundle of hemlock needles and twigs was prepared.

A kitchen lighter was used to ignite and burn. The burning was monitored, and small amounts of additional hemlock. The burn was unsuccessful due to the continued burn and fuel needed.

Fig. 8. Shows the hemlock needles tree branch was unsuccessfully ignited.

In a closed area a piece of A4 size dry white paper used for the study. kitchen lighter was used to burn the paper. The smoke generated was detected.

SEN55 sensor was placed at varying distances from the fire and flam source to investigate the effect on the sensor SEN55 readings. Measurements were taken in both ‘close’ and ‘open’ areas relative to the fire. Sensor was located on a higher level from the sensor level and. The flame and the smoke were moving toward the right side and the sensor was in the left side of the sensor. Close was defined as within 1 meter of the flame, while open’ was defined as 2 meters away.

The impact of wind on sensor readings was evaluated by conducting experiments in a controlled environment with varying wind speeds. A simple wind equation was used to correlate wind speed with changes in particulate matter concentration. Data Acquisition for the Sensor readings were recorded at regular intervals every 1 second for a duration of 3 minutes during each experiment. Baseline readings were also recorded and saved before each fire simulation to establish the ambient air quality stability levels for 5 minutes of each experiment.

E. *Challenges and Solutions*

During the experiment and the study several challenges were encountered during the implementation of the system, and the following solutions were adopted to overcome the challenges.

I2C Communication Issues was exceeded for the Arduino Uno R4 WiFi as it does not have built in pullup resistors for the I2C trace, leading to unstable communication with the SEN55 sensor. This issue was concluded by adding external 1kΩ pullup resistors to the SDA and SCL routing.

Wi-Fi Communication Initial attempts conducted using MQTT and TCP/IP for data transmission were unsuccessful due to compatibility issues and error with the Arduino Uno R4 WiFi. HTTP POST implemented were ultimately used to transmit data to the Raspberry Pi Flask server.

Sensor Calibration the SEN55 sensor required a warm-up period of approximately around 2-5 minutes to stabilize the sensor readings. All experiments were conducted and performed after the sensor had reached its stable steady state.

Material Variability was used for the study. Some materials are red cedar and hemlock and dry leaves for the open area and dry paper for the close area. The red cedar and hemlock were harder to burn, resulting in less observed changes in sensor readings. Dry leaves were used as the primary material for fire experiments due to their rapid combustion, consistent results and the time it took to burst into flames.

F. *Future work*

While the current system focuses on air quality monitoring, the following components will be integrated in the future work:

Vision System with A camera module going to be added to provide visual confirmation of fire events with the volume of the smoke and the fire. This is important for the upcoming study. The vision system will be integrated with the air quality monitoring system to improve detection accuracy and scalability. AI Accelerator ‘Hailo’ as a Machine learning model, is trained to monitor fire events based on sensor data. The Hailo accelerator will be intensely used to deploy these models for real time inference.

Full System Integration with all components ‘air quality monitoring, vision system, and AI accelerator’ will be integrated into a unified solution. This will be the focus of subsequent papers.

III. RESULTS

This section presents the experimental results obtained using the air quality monitoring system for wildfire detection. The data received during the experiments being analyzed to evaluate and study the system performance in detecting smoke and fire events. The results are organized into sections and subsections based on the key research questions addressed in the study. SEN55 has been experimented by their manufacture using Arduino UNO R3 that includes the integrated Pull up resistor. UNO R4 does not have pullup resistor. External resistors need to be added to R4. The reason for using R4 is the WiFi capability that is unavailable with the R3 version. Experience for UNO R4 with WiFi needs to be studied by adding the pullup resistor and comparing the result to the UNO R3. The data collected during the experiment with be investigated in this section.

A. *Adintifing the correct setup*

Identifying the stable setup is crucial for the setup. The selected setup will be used as a baseline for conducting research. To choose effective setup experiments has been processed. The experiment was done on 3.3Vdc and 5Vdc as the power source.

B. *Baseline Sensor Readings*

Before conducting fire experiments, baseline readings were recorded to establish normal environmental conditions. The following average values were observed in a controlled environment without smoke or fire. The results are as follows:

Using Uno R4 WiFi board with external pull up resistors that are connected to the 5Vdc Arduino power source. The results are shown on Fig. 9 and as follows:

PM1.0: 2.70 µg/m³, PM2.5: 2.90 µg/m³, PM4.0: 2.90 µg/m³, PM10.0: 2.90 µg/m³, Humidity:97.75%, Temperature:0.50°C, VOC Index: 102.00, NOx Index: 1.00

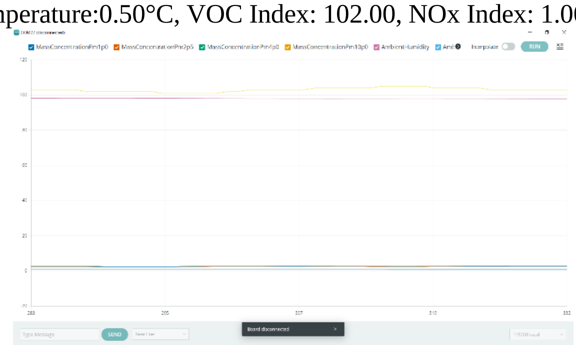

Fig. 9. The figure shows the serial plotter of SEN55 reading of Arduino UNO R4 with WiFi capability, that illustrate the stability of the system after 4 minutes of running under normal conditions. The sensor needs approximately 4 minutes before starting detection for accurate reading.

Using R3 board with internal pull up resistor. The results are shown on Fig. 10 as follows:

PM1.0: 2.20 µg/m³, PM2.5: 2.30 µg/m³, PM4.0: 2.30 µg/m³, PM10.0: 2.30 µg/m³, Humidity:97.75%, Temperature:0.68°C, VOC Index: 98.00, NOx Index: 1.00

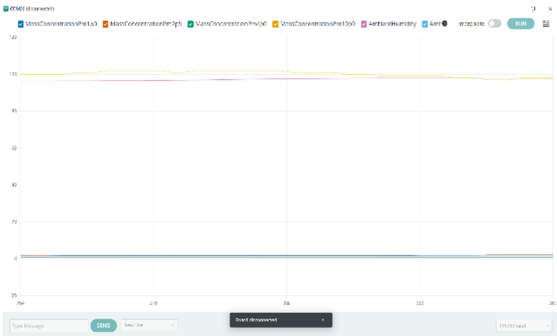

Fig. 10. The figure shows the serial plotter of SEN55 reading of Arduino UNO R3, that illustrate the stability of the system after 4 minutes of running under normal conditions. The sensor needs approximately 4 minutes before starting detection for accurate reading.

These baseline values served as a reference for identifying abnormal conditions during fire experiments.

C. Sensor Response to Smoke and Fire

The system was able to detect smoke as well as fire events was evaluated by burning branches from three types of trees: a tree with dry leaves, red cedar, and hemlock. These materials were chosen to simulate real world wildfire scenarios and conditions, as wildfire often originates from trees and from vegetation. The results of the dry paper are summarized.

Tree with Dry Leaves: The branch from the tree with dry leaves was highly combustible shown in Fig. 11, producing significant changes rapidly in the sensor readings within a second of ignition. The ignition applied two consequences due to burning instantly to all the leaves.

First Peak Values During Combustion:

PM1.0: 2431.50 µg/m³, PM2.5: 2586.70 µg/m³, PM4.0: 2617.80 µg/m³, PM10.0: 2632.30 µg/m³, Humidity: 95.15 %, Temperature: 1.04°C, VOC Index: 157.00, NOx Index: 1.00

The rapid increase in particulate matter concentration and VOC levels demonstrated the system's ability to detect fire and smoke events at an early stage.

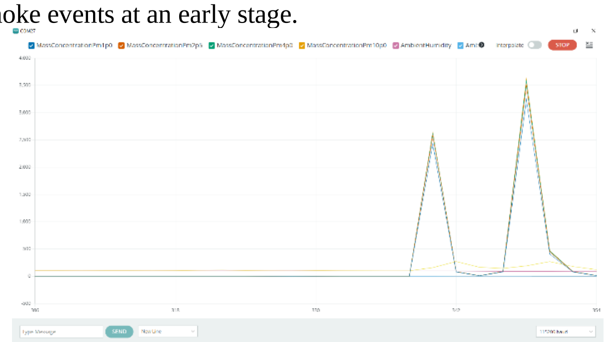

Fig. 11. The figure shows the dry leaves ignited readings during combustion in open areas that are rabidly increased the sensor response of two consequence times of burning using SEN55.

Second Peak Values During Combustion:

PM1.0: 3231.10 µg/m³, PM2.5: 3483.10 µg/m³, PM4.0: 3563.00µg/m³, PM10.0: 3600.30µg/m³, Humidity: 88.58 %, Temperature: 2.16°C, VOC Index: 157.00, NOx Index: 190.00

Red cedar branch was less combustible than the dry leaves, causing in moderate changes in sensor readings. Hemlock is Similar to red cedar, the hemlock branch was less combustible, generating smaller changes in sensor readings.

Dry white paper was in close area was less combustible compare to the fry leaves in the open area. The sensor reading was tremendously increasing and changes in the sensor reading as sowing in Fig.12.

Dry A4 white paper Peak Values During Combustion:

PM1.0: 5826.30 µg/m³, PM2.5: 6553.40 µg/m³, PM4.0: 6553.40 µg/m³, PM10.0: 6553.40 µg/m³, Humidity: 46.52 %, Temperature: 25.72 °C, VOC Index: 368.00, NOx Index: 1.00

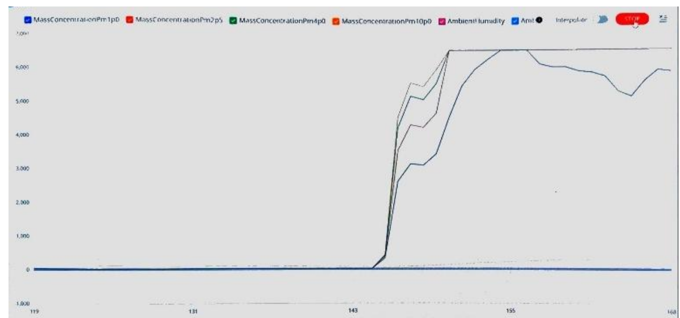

Fig. 12. The figure shows the dry paper ignited readings in close environment during combustion that are exponentially increased the sensor response.

The high rise in particulate matter concentration and VOC levels demonstrated the system's ability to detect fire and smoke events at an early stage. The indoor reading was stable constant during the burning timeframe due to no wind and smoke and the fire concentrating in the surrounding close area.

D. Comparison of Open vs. Closed Environments

Experiments were conducted in both open and closed environments to evaluate the system's performance under different conditions. The results are summarized in Table 1.

TABLE I. Open and closed environments sensor reading evaluation.

Parameter	Open Environment	Closed Environment
PM1.0 (µg/m³)	2431.50	5826.30
PM2.5 (µg/m³)	2586.70	6553.40
PM4.0 (µg/m³)	2617.80	6553.40
PM10.0 (µg/m³)	2632.30	6553.40
Humidity (%)	95.15	46.52
Temperature (°C)	1.4	25.72
VOC Index	157.00	368.00
NOx Index	1.00	1.00

Open Environment Sensor readings were lower compared to the closed environment due to the dispersion of smoke and particulate matter. Closed Environment is higher concentrations of particulate matter and VOC were observed due to the confined space.

E. Data transfer to the core processor

Over the WiFi data that was transferred to the raspberry pi 5 using the flask server. Sensor data was updated every second. The IP address identified during the Arduino code to match the Flask server as well as its port. A combination of sensor readings was sent simultaneously while containing the particulate matter ‘PM1.0, PM2.5, PM4.0, PM10.0’, volatile organic compounds (VOC), nitrogen oxides (NOx), ambient temperature, and relative humidity. The results align with the same results that were collected from the serial and plotted monitors tools in Arduino.

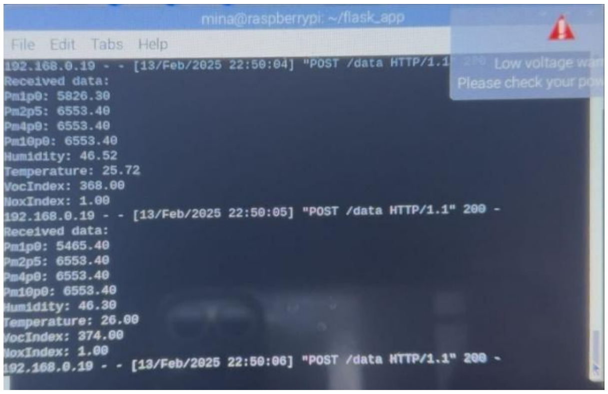

Fig. 13. The figure shows the Data transferred to raspberry pi using HTTP HOST and Flask server. The data was updated every second and matched with the data collected and sent by ESP32.

F. Wind Effect on Sensor Readings

The impact of wind on sensor readings was evaluated by conducting experiments with varying wind speeds. A simple wind equation was used to correlate wind speed with changes in particulate matter concentration. Where:

C = Particulate matter concentration at wind speed
C0 = Initial particulate matter concentration
k = Decay constant
v = Wind speed

TABLE II. Data used to apply to the wind equation.

Scenario	PM2.5 Concentration (µg/m³)	Wind Speed (m/s)	Humidity (%)	Temperature (°C)
Closed Environment ‘Indoor, Burning Paper’	6553.40	0.00	46.52	25.72
Open Environment with Burning ‘Dry Leaves’	2586.70	1.34	95.15	1.04
Open Environment Without Burning ‘Natural’	2.90	1.34	97.75	0.50

Wind equation to accurately predict the observed PM2.5 concentration (2.90 µg/m³) in natural conditions, the wind speed needs to be approximately 11.15 m/s ‘24.9 mph’. This is much higher compared to the measured wind speed ‘7 mph or 3.13 m/s’, indicating that the decay constant k=0.693s/m is not sufficient to explain the observed PM dispersion in natural conditions.

The original decay constant works good for controlled burning scenarios but underestimates the PM dispersion in natural conditions.

For natural conditions with a wind speed of 7 mph ‘3.13 m/s’, a revised decay constant k=2.47s/m is needed to accurately calculate the observed PM2.5 concentration which is 2.90 µg/m³.

This suggests that natural wind patterns and environmental factors significantly affect PM dispersion, and the decay constant

K needs to be adjusted with study under different wind conditions and calculated based on the specific conditions.

G. System Performance

System demonstrated outstanding high sensitivity to smoke and fire events, with the following key observations from the study. Early Detection was very efficient as of the system detected fire events within seconds of ignition fire, as evidenced by the steep increase in particulate matter results concentration reading and VOC levels. Material Variability of the system was highly efficient at detecting fires involving highly combustible materials such as branches with dry leaves.

Environmental Conditions that the study conducted under was detected by the system performed reliably in both open and closed environments, with higher sensitivity in closed spaces.

H. Challenges and Limitations

Material combustibility: branches from red cedar and hemlock trees were harder to burn, causing less pronounced changes in sensor readings.

Wi-Fi Communication Initial attempts to use MQTT and TCP/IP for data transferring were unsuccessful, requiring a switch to HTTP POST requests with Flask server.

Sensor Warm-Up Time for the SEN55 sensor required a warm-up period of 2-5 minutes to stabilize its readings, delaying the start of experiments. Understanding the warm-up timeframe is very important to ensure accurate reading results.

I. Upcoming Work

The following components will be integrated and added to the future work to enhance the system’s capabilities and scalability.
Vision System with a camera module is in the development phase and will be added to provide visual confirmation of fire events.
AI Accelerator ‘Hailo’ with Machine learning models will become trained to predict fire events based on sensor data results.
Full System going to be Integrated with all components into a unified solution for real-time wildfire detection.

The results demonstrate the effectiveness and importance of the air quality monitoring system for early wildfire detection. The system’s performance is able to detect rapid and quick changes in particulate matter concentration and VOC levels allowing it to be a favorable tool for fire prevention and response. The use of tree branches ‘dry leaves, red cedar, and hemlock’ in the experiments underline the real-world applicability of the system, as wildfire often originates from trees and vegetation. On the other hand, the variability in material combustibility and environmental conditions highlights the need for further optimization and integration with additional devices and systems.

IV. CONCLUSIONS AND FUTURE WORK

This study presents a scalable, AI-powered wildfire detection system integrating real-time air quality monitoring and vision-based fire verification. The system's was able to detect steep changes in particulate matter concentration with different matter sizes as well as VOC and NOX levels. It is a promising technique for wildfire prevention and illumination. Controlled experiments using tree branches such as dry leaves, red cedar, hemlock and dry paper have shown that the system can identify and detect the fire events within seconds of ignition of the fire, with tremendous changes in sensor readings observed in both open and closed environments with different conditions such as wind and temperature.

The use of the SEN55 air quality sensor, combined with the Arduino Uno R4 WiFi and Raspberry Pi 5, provides a robust design and scalable solution for real time data collection, transferring and processing. The integration of an AI accelerator 'Hailo' and vision-based systems, currently under development and study, will further enhance the system's capabilities by allow real time machine learning inference and visual system confirmation of fire different events

The challenges experienced during the execution, involves I2C communication error and WiFi data transferring limitations, were effectively resolved with providing valuable insights for future optimization. The investigation of different data transmission protocols, culminated with the successful implementation of HTTP POST with a Flask server, showing the magnitude of selecting appropriate communication protocols for real time sensor data transmission.

Future work is going to focus on integrating a vision system and an AI accelerator powered by 'Hailo' to enhance the system's ability and scalability to detect smoke and fire. The vision machine will provide visual confirmation of fire events occurs, while the AI accelerator will enable real time machine learning inference for fire confirmation and detection. The consolidation of these components open a new realm of comprehensive, unified solution for wildfire monitoring and detection response.

The results of the study contribute to the growing body of research of early wildfire control and detection systems. That implemented by providing a robust and efficient methodology for monitoring critical environmental factors, this system offers a signified research with successful tools for mitigating the devastating impacts of wildfires on the human being and the earth in a bigger scale.

ACKNOWLEDGMENT

This research is supported by NASA Connecticut Space Grant Consortium under Faculty Research Grant, PTE Federal Award No.: 80NSSC20M0129, grant period: 7/1/2024-05/31/2025. The authors are grateful for support from NASA CT Space Grant Consortium.

REFERENCES

[1] CNN News https://www.cnn.com/us/maps-damage-los-angeles-wildfires-dg/index.html ‘accessed on 2 March 2025’
[2] California Department of Forestry and Fire Protection ‘CAL FIRE’ of 2020 https://www.fire.ca.gov/incidents/2020 ‘accessed on 2 March 2025’
[3] California Department of Forestry and Fire Protection ‘CAL FIRE’ 2024 https://www.fire.ca.gov/incidents/2024 ‘accessed on 2 March 2025’
[4] NASA: Fire Information for Resource Management System https://firms.modaps.eosdis.nasa.gov/ . ‘accessed on 2 March 2025’
[5] Sensirion: Environmental sensor https://sensirion.com/products/catalog/SEN55
[6] Reisen, F., Meyer, C.P., Weston, C.J. and Volkova, L., 2018. Ground-based field measurements of PM2. 5 emission factors from flaming and smoldering combustion in eucalypt forests. *Journal of Geophysical Research: Atmospheres*, *123*(15), pp.8301-8314.
[7] Arduino: UNO R4 WiFi https://docs.arduino.cc/hardware/uno-r4-wifi/
[8] Raspberry pi: raspberry pi 5 https://www.raspberrypi.com/products/raspberry-pi-5/
[9] HAILO: Edge AI https://hailo.ai/
[10] Arduino Uno R4 layout: https://docs.arduino.cc/tutorials/uno-r4-wifi/cheat-sheet/
[11] Broadcom BCM2712 documents https://www.broadcom.com/products/embedded-and-networking-processors/communications/bcm58712
[12] HTTP POST information https://en.wikipedia.org/wiki/POST_(HTTP)