Remote Sensing for Forest Fires

Design Document

UBC Cloud Innovation Centre

Capstone Team CG-23

Mark Lee, Austin Li, Ethan McKeen, Gurjot Sandhu, Christy Zhang

2024-04-30

Table of Contents

List of Figures

List of Tables

1. Project Context

The UBC Cloud Innovation Centre (CIC) is dedicated to fostering innovative solutions that address real-world challenges, with a mission to create a positive impact on British Columbia, Canada, and the global community [1]. Collaborating closely with Amazon Web Services (AWS), UBC CIC focuses on developing shareable, repeatable and sustainable cloud technology solutions.

British Columbia and Canada have recently been experiencing increasingly frequent and severe wildfires, exacerbated by factors such as climate change, prolonged droughts and an accumulation of dry vegetation. According to data provided by the BC provincial government, the current 10-year average is 1,483 wildfires a year and annual damages costing $316.9 million, with both wildfire count and damage costs increasing yearly [2]. Furthermore, these fires pose a significant risk to human lives, property, biodiversity, and the overall ecological health of the region.

This project aims at early prediction and detection of fires. By forecasting the likelihood of fire outbreaks in a given region, it has the capacity to alert individuals planning to travel there, allowing them to postpone their trips and thereby prevent potential injuries and economic losses. Furthermore, the project's fire detection and monitoring systems provide support to firefighters in resource allocation and enable swift responses to emergency situations.

The effective execution of the "Forest Fire Remote Sensing" initiative provides clients and end-users with the chance to utilize an adaptable, open-source solution. This empowers municipalities and small business proprietors to proactively safeguard lives and assets from the destructive consequences of forest fires. This project incorporates cutting-edge early detection technologies, including artificial intelligence, ensuring prompt responses and enabling cost-efficient prevention methods. Through our web-based application and alerts, it reduces risks for both workers and residents in high-risk areas. Moreover, it equips forest firefighters with better tools to manage and prevent the spread of forest fires, thereby minimizing the risk of future outbreaks.

2. High-Level Design

The client's organization faces the challenge of early prediction and detection of wildfires to mitigate their devastating impact. To address this challenge, as seen below in figure 1, this issue can be addressed utilizing a solution consisting of four key functional blocks. These blocks include: intaking data relevant to the detection and prediction of forest fires, storing that data, using the stored data to make calculations and predictions regarding the fire, and relaying this crucial information back to the users. Note the colours of the blocks as the block diagrams later in the document will have matching colours that correspond with how the team will address each block in the function block diagram.

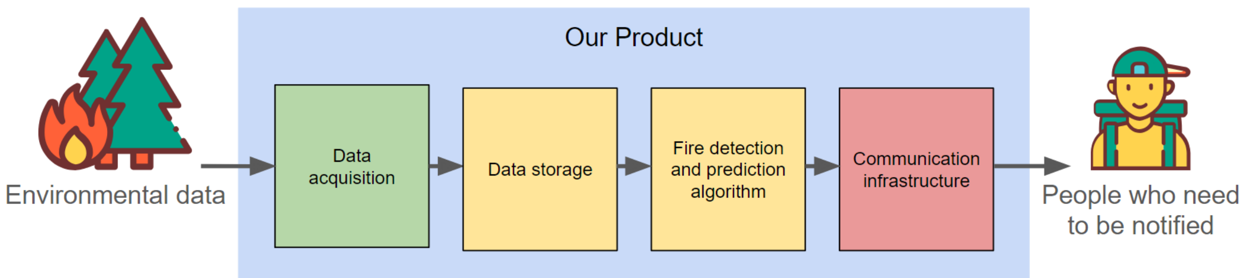


Figure 1: Functional Block Diagram

Our proposed solution involves the development of a deployable prototype that exploits IoT sensors to collect near real-time data in forests and satellite data to help make informed predictions of high risk areas. This data will be stored and processed on a web server where the satellite and sensor data can be cross referenced and used to make risk assessments of potential fire outbreaks. This information will then be made publicly available through a web application for users to view. Important alerts will also be sent to the user if concerning fire risk levels are assessed. The IoT sensor data will reach this web server from remote areas beyond the coverage of wifi with the help of a long range radio frequency gateway. Our solution can be seen below in figure 2. As previously mentioned the colours of figure 2 correspond with the functional block diagram.

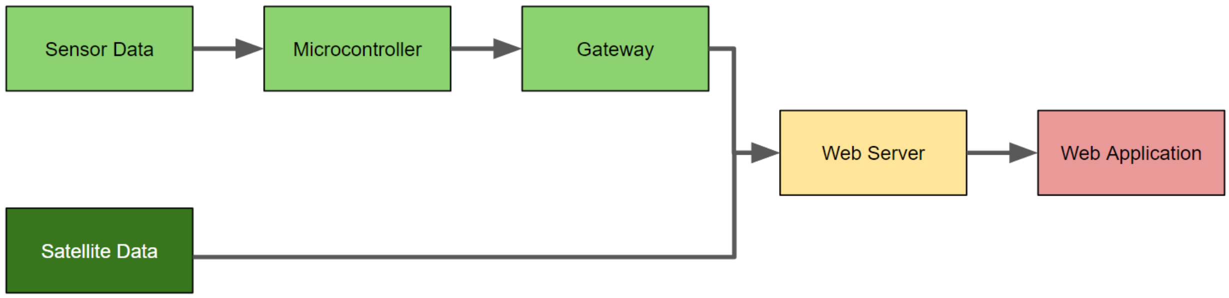


Figure 2: High Level Decomposition

In this decomposition the lighter green blocks indicate the physical device that is intended to be deployed in remote areas, the darker green block indicates the secondary means of intaking data in the form of satellite data. The yellow and red blocks indicate the web implementation of the solution. To maintain clarity and consistency throughout this document, the term “device” will henceforth refer to the physical deployable device as indicated by the lighter green blocks in figure 2.

3. High Level Choices

In the realm of wildfire detection and prediction systems, the method of collecting data is an important decision. Such methods need to be capable of collecting data from remote areas beyond the coverage of wifi and relay this info to a web server. The collected data should be instrumental in predicting and detecting wildfires allowing for early prevention and rapid response times. Potential avenues for data collection include IoT sensors, satellite imaging, and an ensemble of both approaches. This strategic decision significantly shapes not only the type of data acquired but also the frequency at which it is gathered. The main benefits and limitations of IoT sensors and satellite imaging are outlined below.

IoT Sensors	
Benefits	Limitations
• Early Detection • Near-Real Time Monitoring • Remote Sensing	• Cost • Coverage • Forest Fire Prediction Capabilities • Power Supply • Maintenance • False Alarms

Satellite Imaging	
Benefits	Limitations
• Wide Coverage • Forest Fire Prediction Capabilities • Data Fusion • Cost-Effective	• Resolution • Cloud Cover • Sensitivity • Latency • Interference • Interpretation • False Alarms

Table 1: IoT Sensor and Satellite Imaging Comparison

3.1. IoT Sensors

3.1.1. Benefits

Early Detection: Fast response time from the device and continuous monitoring leads to earlier detection of fire and quicker response times.

Near-Real Time Monitoring: The sensors offer near-real time monitoring technology which allows users to track the progression of a fire and make informed decisions promptly. This applies to functional requirement 4.1.4.

Remote Sensing: With the use of technologies such as LoRa, 5G, and LTE these sensors can monitor remote areas outside of wifi range, providing coverage in regions that may not be easily accessible for manual monitoring [3]. This applies to non functional requirement 4.2.3.

3.1.2. Limitations

Cost: More costly than satellite imaging because all hardware parts will have to be purchased and assembled. IoT sensors also have a higher time cost due to the design and production of constrthe physical device. This applies to constraint 5.1.

Coverage: Increasing the area monitored by the sensors requires buying multiple devices. It is difficult to give an accurate effective range of a single device as it may differ due to various environmental conditions. While the use of LoRa, 5G, and LTE greatly increases their effective range, this range is still not nearly as large as satellite imaging. This applies to non functional requirement 4.2.4 and functional requirement 4.1.5.

Forest Fire Prediction: There are some types of “fuel data” that IoT sensors struggle to measure such as the health of the trees in a forest [4]. This type of data helps forest fire prediction and assessing high risk areas. Also the complexity of the device and associated cost escalates with the desire to collect a broader range of data as the purchase of additional/diverse sensors is required. This applies to constraint 5.8.

Maintenance: IoT sensors may be subject to technical failures or environmental conditions that affect their reliability. Regular maintenance and monitoring are essential to ensure accurate and consistent performance. This applies to non functional requirement 4.1.3.

False Alarms: Environmental factors such as smoke from controlled burns, dust, or other particles can trigger false alarms, leading to unnecessary responses and resource allocation.

3.2. Satellite Imaging

3.2.1. Benefits

Wide Coverage: Effective for detecting fires in remote, unpopulated regions, where conventional fire monitoring is less intensive. This applies to non functional requirement 4.2.4 and functional requirement 4.1.7.

Forest Fire Prediction: Can assess which areas are high risk and how likely a fire is to spread using “fuel data” [4]. This includes: wind direction, rain, whether trees in an area are healthy or the area is dry/in a drought, the beetle population in an area[1], whether the trees are standing or there are lots of fallen down trees[2]. These factors would be very challenging for IoT sensors to accurately detect. This applies to constraint 5.8.

Data Fusion: Satellite imaging can be easily combined with other sources, such as weather data, topography, and historical fire patterns which can improve predictive models. This fusion of data helps in better understanding fire behavior and risk assessment.

Cost-Effective: Can be very cost-effective as the implementation of this solution would be entirely software based. This applies to constraint 5.1.

3.2.2. Limitations

Resolution Limitations: According to the Canadian WildFire Information System (CWFIS), the actual size of the actively burning area cannot be determined from satellite imagery. “A 1-km² hotspot pixel[3] may represent a fire as small as 100 m². In addition, an intense fire covering an area less than 1 km² may actually show up as a cluster of several hotspot pixels. This is the result of the varying size and spatial overlap of the raw, unprojected pixels” [5].

---

[1] Beetles affect the likelihood of a wildfire because they can kill trees creating fuel for the fire and they can accelerate the drying process of trees.

[2] Standing healthy trees are much harder to ignite than a fallen tree. Fallen trees increase the risk of wildfire spread.

[3] A hotspot pixel refers to a specific pixel on a satellite image that indicates an area of intense heat or fire activity. The size of this pixel is standardized to 1 km² for the CWFIS [5].

Cloud Cover: Weather conditions, especially cloud cover, can obstruct satellite views. Clouds can limit the frequency of data acquisition and affect the reliability of continuous monitoring.The algorithms cannot detect fires through thick cloud or smoke. A large fire may therefore go undetected for several days and then appear or reappear later; a small fire may burn and die out without ever being detected.

Limited Sensitivity: Satellites primarily detect the heat and smoke produced by fires. Some types of fires, such as slow smoldering fires, may not produce significant heat or smoke, making them harder to detect.

Data Latency: The time lapse between satellite image acquisition and image distribution on the CWFIS site is between 1 hour and 7 hours, depending on the sensor and processing time, and ECMWF is every 12 hours. NASA's Land, Atmosphere Near real-time Capability for EOS (LANCE) provides imagery within 3-5 hours after observation. This high latency is an issue if early detection and fast response is a priority.

Interference: Radiofrequency interference or other technical issues can affect the quality and reliability of satellite data.

Data Interpretation: The interpretation of satellite data could be considered outside our scope as we are not Atmospheric Science (ATSC) majors and its integration with other information sources can be complex, requiring skilled personnel for effective use. This could extend the time to complete the project beyond our constraint of 2 school terms. This applies to constraint 5.4.

False Alarms: Satellite data can sometimes produce false alarms due to factors like industrial activities, controlled burns, or non-fire heat sources.

3.3. Ensemble Approach

Our team has strategically chosen to integrate both IoT sensors and satellite imaging for enhanced wildfire detection and prediction capabilities. By leveraging IoT sensors, we achieve near-real-time monitoring, crucial for early detection and swift response times. However, these sensors are limited in their ability to collect comprehensive data, particularly regarding important factors like fuel data essential for wildfire prediction. To address this limitation, we incorporate satellite imaging, which provides a fully software-based implementation. Satellite imaging excels in capturing fuel data, enabling more accurate assessments of high-risk areas. Although it faces data latency challenges, this approach combines the strengths of both methods, resulting in early detection and improved predictive analytics for more effective forest management strategies.

Ensemble Approach	
Benefits	Limitations
• Early Detection • Near-Real Time Monitoring • Remote Sensing • Wide Coverage • Forest Fire Prediction Capabilities	• Cost • Power Supply • Maintenance • False Alarms

Table 2: Ensemble Approach Evaluation

3.4. Fire Weather Index System

In addition to the method of collecting fire data, it is important to cover how our team has chosen to dictate what data is relevant. Our team has opted to use the Fire Weather Index (FWI) System to calculate the risk of a fire. This was chosen as the FWI System is the main risk assessment system utilized by the Canadian Government and the Canadian Wildland Fire Information System (CWFIS) which “monitors fire danger conditions and fire occurrence across Canada” daily by creating fire behavior and fire weather maps [6]. As seen below in figure 3, the FWI System utilizes 6 components that account for the effect of moisture levels and weather conditions on the risk of a fire. The system uses temperature, relative humidity, wind speed, and 24-hour precipitation as inputs into these six components. These components offer numerical ratings indicating the relative risk of wildland fire and their functionality is outlined below.

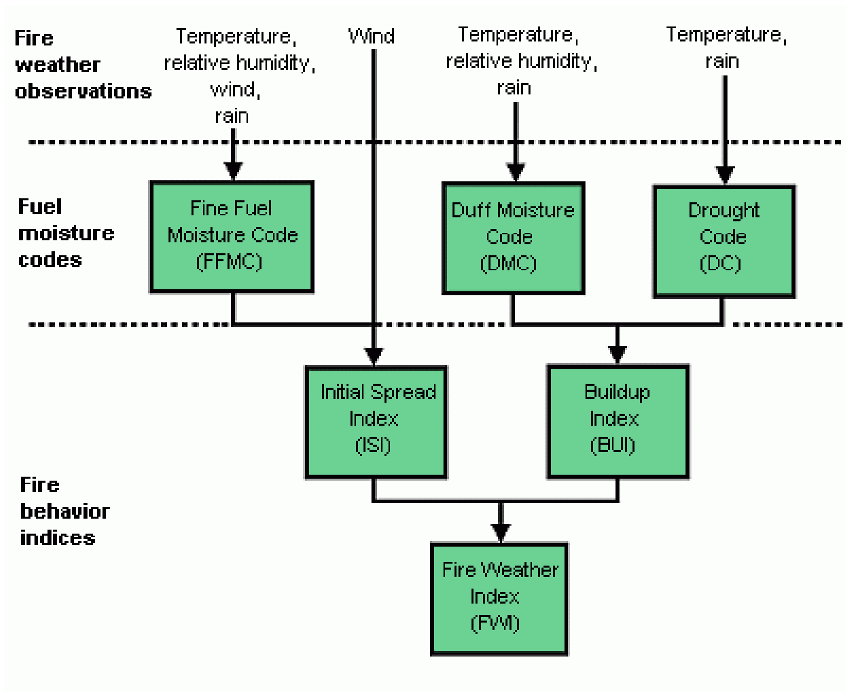


Figure 3: Fire Weather Index System [7]

3.4.1 Fine Fuel Moisture Code

The Fine Fuel Moisture Code uses the temperature, relative humidity, wind speed, and rainfall to calculate a numeric rating of the moisture content of fuels on the surface level of the soil and other fine fuels[4]. This numeric rating indicates the ease of ignition and flammability of fine fuels [7].

[4] "Fine fuels - Fuels that ignite readily and are consumed rapidly by fire (e.g. cured grass, fallen leaves, needles, small twigs). Dead, fine fuels also dry very quickly" [8].

3.4.2 Duff Moisture Code

The Duff Moisture Code uses the temperature, relative humidity, and rainfall to calculate a numeric rating of the moisture content at a moderate soil depth. This depth is above the mineral soil but below the litter[5] layer. This numeric rating indicates the fuel consumption at these moderate depths and within medium sized wood objects [8][7].

3.4.3 Drought Code

The Drought Code uses the temperature and rainfall to calculate a numeric rating of moisture content at deep compact soil layers. This rating indicates the seasonal drought impacts on forest fuels, as well as the extent of smoldering within deep soil levels and sizable logs [7].

3.4.4 Initial Spread Index

The Initial Spread Index uses the wind speed and the rating given by the Fine Fuel Moisture Code to give a numeric rating indicating the rate at which the fire is expected to spread [7].

3.4.5 Buildup Index

The Buildup Index uses the ratings given by the Duff Moisture Code and the Drought Code to give a numeric rating indicating the total amount of fuel readily available for a fire to consume [7].

3.4.6 Fire Weather Index

The Fire Weather Index is the output of the entire system. It uses the Initial Spread Index and the Buildup Index to give a numerical rating indicating the risk of a fire occurring in the forested area [7]. This fire danger rating ranges from 0 to 5 where 0 indicates a very low risk of fire and 5 indicates an extremely high risk of fire as seen in figure 4.

---

[5] "Litter - The uppermost part of the forest floor consisting of freshly fallen or slightly decomposed organic materials" [8].

Fire danger class number	Fire danger class name
0	Very low
1	Low
2	Moderate
3	High
4	Very high
5	Extreme

Figure 4: Numerical Rating Range for FWI [9]

4. Low Level Choices

With the decision to integrate both IoT sensors and satellite imaging, each individual component of the device must be selected. These components include: the data transmission system, the microcontroller, IoT sensors, power supply, satellite data source, machine learning algorithm, web server, and web application. It is crucial that these decisions enable the device to function as intended without imposing an overwhelming workload that might hinder the team from completing the project within the specified timeframe.

4.1. Long Range Data Transmission

Data Transmission plays a significant role in the design as the device is required to have the capability to monitor remote areas outside of wifi coverage. A signal must be sent from the remote IoT sensor monitoring device to a web server. The team looked at multiple methods of accomplishing this including LTE, 5G, and LoRaWAN. Long Term Evolution (LTE) and 5G both have a significantly shorter range than Long Range Wide Area Network (LoRaWAN). Using 5G would provide a large bandwidth (1Ghz - 6Ghz) and high data rate but higher frequency leads to higher propagation loss and shorter transmission distance [10]. As seen in figure 5, LoRa provides both the range and bandwidth required for our application. The near-real time rate at which our data will be transferred does not need this high bandwidth and high data rate.

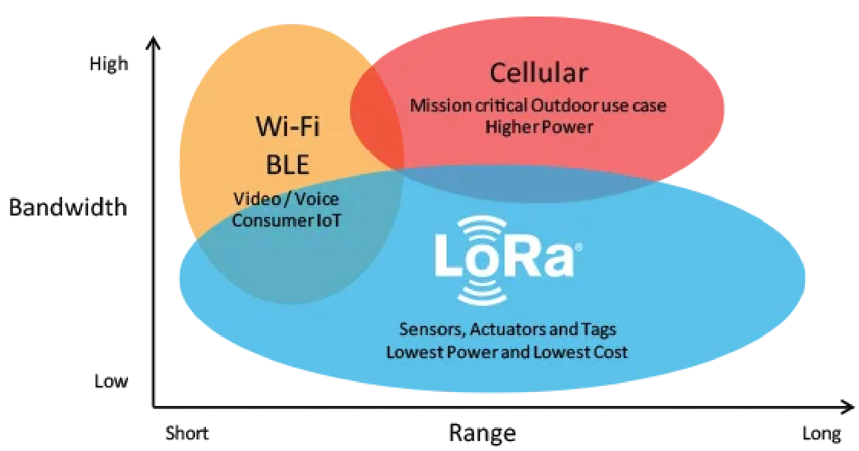


Figure 5: LoRa Comparison Chart [3]

Figure 6: SenseCAP M2 Multi-Platform

Our team chose LoRa as the preferred technology for long-range data transmission. In particular, the team chose the SenseCAP M2 Multi-Platform LoRaWAN Indoor Gateway (figure 6) [11]. Selecting a LoRaWAN gateway implies that the microcontroller of the sensor device needs to be equipped with a LoRa radio module. LoRaWAN was chosen because it provides the longest data transmission range (over 10km) among our options and it is highly scalable as it uses a wifi model. This wifi model means that with the purchase of a single gateway, similar to a wifi router, multiple devices can be connected and a large network can easily be built. Also LoRaWAN operates on a free public spectrum so no frequency license is required (These ISM bands include EU868MHz, AS923MHz, US915MHz). As seen below in figure 7, a remote LoRa sensor device will send its data to a LoRaWAN gateway. This gateway will be inside wifi coverage allowing for direct communication with The Things Network[6] (TTN) where the web server and web application can view the data. Among LoRa gateways there are two options: single channel and multi-channel. Unlike multi-channel, single channel gateways can only talk to one end device at a time but they are cheaper. Our team chose a multichannel gateway as we want to grant end users the ability to build large networks of devices to increase the amount of area monitored.

[6] The Things Network - TTN is a LoRaWAN Network Server which manages applications, end devices and gateways [12].

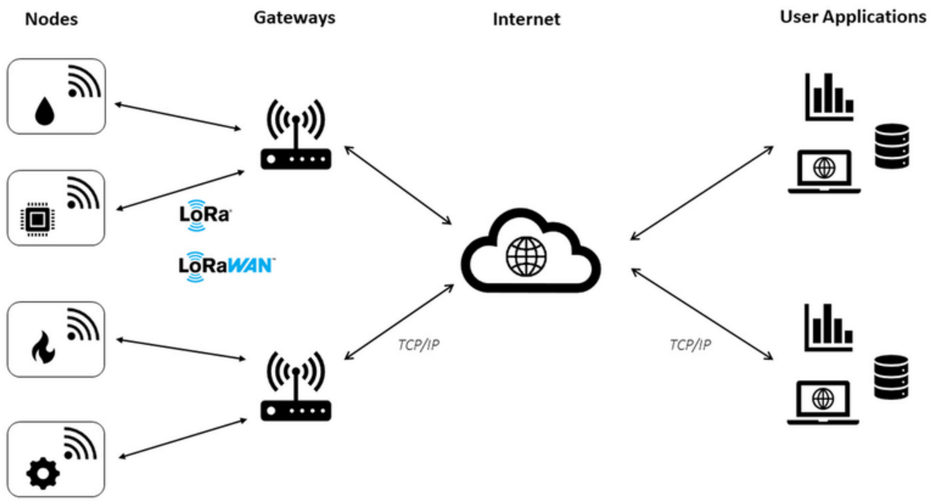


Figure 7: LoRa Data Transmission [13]

The team chose a EU868 ISM band gateway which operates in the 863-870 MHz frequency range. Although the US915 ISM band has a larger number of channels, providing more flexibility in avoiding interference and accommodating a higher number of devices, the lower frequency of EU868 provides slightly better range and penetration through obstacles.

4.2. Microcontroller

The microcontroller is a crucial component in the sensor device as it will transmit the data collected by the sensors to the gateway. Since the preferred method of data transmission is LoRa, the microcontroller must come equipped with a LoRa radio. The first decision the team made was choosing between a standalone microcontroller chip, using a single-board computer like Raspberry Pi, choosing pre-built sensor and microcontroller modules like the options offered by SenseCap, or using a development board with the MCU pre-installed. A standalone microcontroller chip would require the team to design and print our own PCB which would increase and complicate the time to develop a working sensor device. Opting for a single-board computer, such as the Raspberry Pi, would be excessive for our application, given its inclusion of a full-fledged operating system. This means the size and power requirements would be much larger. Another option was to use pre-built sensor and microcontroller modules that companies like SenseCAP offer [14]. This would eliminate the time to build a sensor device but it greatly limits the types of sensors available. In turn, this also limits the types of data collected and further hinders its ability to collect comprehensive data essential for wildfire prediction and detection.

Microcontroller	Cost	Ease of Implementation	Limits Other Aspects of the Design	Power Requirement	Size	Chosen?
Standalone MCU chip (Figure 9)	Low	Low	No	Low	Low	Yes (for final product)
Raspberry Pi (Appendix A.1)	High	High	No	High	High	No
SenseCAP (Appendix A.2)	High	High	Yes	Low	High	No
MCU and Dev Board (Figure 8)	Low	High	No	Low	Medium	Yes (for prototyping)

Table 3: Microcontroller Decision Table

Our team decided that using a development board featuring a pre-installed microcontroller was the best solution for prototyping and designing our own PCB was the best solution for our final product. A large bulk of the hardware design process involves writing and debugging the firmware. By purchasing a pre-built MCU development board the team can start the firmware design process immediately without needing to wait for a finished PCB to be designed and manufactured. This solution offers flexibility in both sensors and implementation. We chose the Wio-E5 mini Dev board (figure 8) because it came with the same standalone STM32 MCU (figure 9) that we planned to use on our PCB [15][16]. The Wio-E5 is also inexpensive with a low power draw and LoraWan capabilities. Opting for a development board with a pre-constructed PCB streamlines the process, as the design of the final product's PCB can have a reference of a working prototype while being done in conjunction with the firmware design. More details regarding the design of this PCB can be seen in section 5.3.

4.3. IoT Sensors

The team encountered a plethora of options when considering IoT sensors. After careful deliberation, we decided to incorporate temperature, humidity, and gas sensors into our project. The sensor must have an I2C output for compatibility with the microcontroller and because it decreases the quantity of connections among devices, simplifying system design and minimizing signal interference. This also makes it easier to increase the number of sensors in the design. Measuring temperature, humidity, and gas are crucial for wildfire detection and prediction due to their significant impact on the environmental conditions that influence the likelihood and behavior of wildfires. Temperature and humidity are key components of the Fire Weather Index as they are the inputs into the Fine Fuel Moisture Code, Duff Moisture Code, and the Drought Code[17]. Elevated temperatures and low humidity levels contribute to drier vegetation, creating favorable conditions for the rapid spread of wildfires. Low humidity levels increase the evaporation rates of moisture from vegetation and soil. Dry vegetation becomes more susceptible to ignition, and the overall landscape becomes more conducive to the ignition and spread of wildfires. These temperature and humidity sensors, along with the wind and rain data collected from satellites, form the crucial building blocks of the Fire Weather Index System. Gas sensor data is also extremely crucial as they can detect specific gasses emitted during the initial smoldering stages of a forest fire, allowing for early detection before the fire becomes large and uncontrollable.

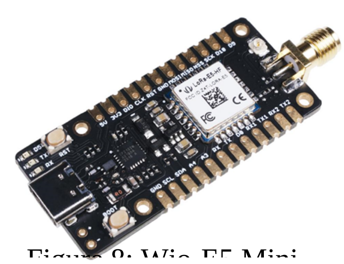


Figure 8: Wio-E5 Mini

The team chose the BME688 sensor (figure 10) which comes with a gas, temperature, humidity, and pressure sensor[18]. This specific sensor was chosen as the gas sensor can detect both carbon dioxide (CO2) and volatile organic compounds (VOC's)[7] which are crucial for determining the presence of a fire. The BME688 sensor is rated to detect a 2 x 2m fire from 100m away. Note that the effective range of these IoT sensors fluctuate greatly as their range is highly dependent on many environmental factors such as wind direction, the intensity of the fire, and the density of the forest. Considering that a dev board was chosen for the MCU to begin firmware design early, the same was done for the BME688 sensor. The BME688 Eval Board, as seen in figure 11, was utilized during prototyping to ensure that firmware development began early in conjunction with the PCB design which utilized the standalone BME688 sensor chip. Bosch Sensortech has written a pre-compiled library called BSEC which was utilized for processing the BME688 outputs [19].

Figure 10:

Figure 11: Eval Board

It is worth noting that the team originally chose DFRobot's SEN0385 SHT-31 temperature and humidity sensor (Appendix A.3). This particular sensor was selected for its cost-effectiveness and waterproof design. However, the ability to detect gasses was proven to be much more beneficial than originally anticipated and this sensor has since been removed from the design.

Other sensors were considered such as Amphenol Telaire's T9602-3-D temperature and humidity sensor which had a high price point and a suboptimal operating range. We also considered the eval board sensor using the BME680 which has the same operating ranges as BME688 at a cheaper price but it lacked the ability to detect carbon dioxide (CO2).

IoT Sensor	Cost	Temperature	Humidity	CO2/VOC	I2C	Chosen?

[7] Volatile organic compounds - "VOCs are chemicals that can be emitted from a variety of products and processes, impacting air quality, public health, and climate" [20]. Fires produce various volatile organic compounds (such as benzene) as byproducts of combustion.

	(CAD)	Range (°C)	Range (%RH)	Detection	Output	
DFRobot SEN0385 (Appendix A.3)	$30.21	-40 ~ 125	0 ~ 100	No	Yes	No
BME688 Eval Board Sensor (figure 10)	$31.95	-40 ~ 85	0 ~ 100	Yes	Yes	Yes
Amphenol Telaire T9602-3-D (Appendix A.4)	$75.93	-20 ~ 70	0 ~ 95	No	Yes	No
BME680 Eval Board Sensor (Appendix A.5)	$26.95	-40 ~ 85	0 ~ 100	Only VOC	Yes	No

Table 4: IoT Sensor Decision Table

4.4. Satellite Data

The choice over which open source satellite data set to use is crucial for collecting the types of data that IoT sensors struggle with. As seen in section 3.4 in figure 3, the major components of the Fire Weather Index System not addressed by IoT sensors include wind and rain observations. Other important factors, as previously discussed in the benefits of satellite imaging section, include: whether trees in an area are healthy or the area is dry/in a drought, the beetle population in an area, and whether the trees are standing or there are lots of fallen down trees. While many options address this, the team opted to use OpenWeather [21]. The OpenWeather API is a service that provides access to both raw weather and forecast data as well as Fire Weather Index data. It offers various endpoints to query weather information based on location, time, and other parameters, enabling our applications to display accurate weather updates to users. It offers a free plan that allows our API key to be called up to 60 times a minute (1000 per day) which is more than enough for our application as our near-real time monitoring system will update roughly every 15 minutes. Furthermore, the team has previous experience with this platform which mitigates the risk of potential delays and enhances the likelihood of completing the project within the established timeframe.

Alternatives were considered like Earthdata.Nasa which provides access to a wide range of Earth science data, including satellite data, through its various portals and tools. It offers a wealth of information but requires significant familiarity with the interface and understanding of how to navigate the available datasets. OpenWeather and Open-Meteo, on the other hand, primarily focuses on weather forecasting and provides weather data and API services that are relatively easy to integrate into applications. While it may not offer the same breadth of satellite data as Earthdata NASA, it can be more straightforward for users looking specifically for weather information. Although Open-Meteo allows more API calls than Open weather, it does not provide FWI calculations which is crucial for streamlining our application.

Satellite Data Service	Cost	Usage	Knowledge Demand	Forecast Data	FWI Data	Chosen?
OpenWeather	Free	1,000 API calls per day	Low	Yes	Yes	Yes
Earthdata.Nasa	Free	Unlimited	High	Yes	Yes	No
Open-Meteo	Free	10,000 API	Low	Yes	No	No

		calls per day				

Table 5: Satellite Data Service Decision Table

4.5. Power Supply

The power supply unit (PSU) is a crucial component in the design of the device as it supplies power to both the microcontroller and the IoT sensors. Considering the requirements of the IoT sensors and the microcontroller, the power supply must produce at least 111μA at 3.3V. The options that the team analyzed for the PSU were solar panels and lithium ion batteries (both rechargeable and non-rechargeable). Non-rechargeable lithium ion batteries are the cheapest up front option but require frequent maintenance as they will need to be replaced when drained. This is a large concern considering that the device is required to function in remote locations that are not necessarily easy to access. Solar panels are a more reliable solution in remote locations due to its self-charging technology as frequent maintenance is not required. However, solar panels have a large cost in regard to both time and money. This solution requires more time and effort to implement than lithium ion batteries due to the time cost of design and production. This means that implementing solar panels poses a high risk of delaying the project's completion past the 2 school term time period (per constraint 5.4). As a result the team opted for rechargeable lithium ion batteries as they are easy to implement and, due to its rechargeability, the maintenance is better than the non-rechargeable alternative. The team opted to use a battery capacity of 1200mAh as the board uses a switching regulator that preserves power with 92% efficiency. So the current consumption of the 3.6V battery will be 1/0.92 * 111μA * 3.3/3.6 = 110μA. Therefore, the battery would last roughly 1200 e^{-3}/110 e^{-6} = 10810 hours or 1.2 years (per non functional requirement 4.2.12). The PCB was designed with a micro-USB port which will charge the battery when plugged in.

Power Supply	Cost	Ease of	Maintenance	Chosen?

		Implementation		
Solar Panel	High	Low	Low	No
Non rechargeable lithium ion batteries	Low	High	High	No
Rechargeable lithium ion batteries	Low	High	Medium	Yes

Table 6: Power Supply Decision Table

4.6. Enclosure

The enclosure for all of the components is the key part of the design that ensures its waterproof capability. The enclosure must fit the microcontroller, sensors, and power supply into a compact space. A smaller enclosure is more beneficial as this makes mounting the device in remote locations of the forest much easier. There must be some port on the enclosure for connecting an antenna and there must be some opening allowing for the conduction of gas without sacrificing its waterproofing. The two manufacturing methods considered for the enclosure were 3D modeling and printing the enclosure or purchasing a pre-manufactured enclosure. Considering the most space efficient design for the enclosure would be a rectangular box it is very easy to find pre-manufactured waterproof enclosures with an optimal size and shape. The higher customizability of a 3D modeled enclosure was not deemed valuable as the simple design needed for the enclosure meant there was a plethora of options offered by pre-manufactured enclosures. 3D modeling would also introduce a large time cost from modeling and printing.

Enclosure	Cost	Ease of	How well the	Chosen?

		Implementation	components will fit	
3D modeled	Low	Medium	High	No
Pre manufactured	Low	High	High	Yes

Table 7: Procurement of Enclosure Decision Table

Enclosure	Cost	Compactness	How well the components will fit	Chosen?
AN-00P	Low	Medium	High	No
AN-20P	Low	Medium	High	No
AN-02P	Low	High	High	Yes

Table 8: Type of Enclosure Decision Table

The team opted for Polycase's AN-02P Diecast Aluminum NEMA Enclosure, as shown in figure 12 [9]. The dimensions of the enclosure are 115 x 65 x 30 mm. The enclosure's metal build offers high durability which is important considering the product will be deployed outside in the wilderness. This enclosure is IP67 rated as it comes with a rubber gasket to ensure the lid is water tight. Two have been drilled into the front for an antenna cable connector and a vent to allow gas to pass through. Both of these are waterproof and can be seen in Appendix A.7. and A.8.

Figure 12: AN-02P Diecast Aluminum

4.7. Web Server

Web server choice is crucial to our project since it provides the functionality to store and process data. Our team examines three possible solutions: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. In Table 8, we summarize the benefits and limitations of all three options and decided that AWS best fits our purposes. We choose AWS considering it offers a wide range of complementary services, enabling us to easily integrate our IoT sensors. In addition, AWS has a global network of data centers and redundant systems to guarantee reliable uptime for our application and be capable of adapting to changing workloads. Their large global footprint is also important to minimize latency and handle growing demands if the product is applied in other areas or countries. Furthermore, AWS has the largest market share in the cloud services sector and is expected to continue receiving substantial maintenance and support in the foreseeable future. In conclusion, AWS offers essential services to support our project, enabling us to fulfill our client's requirements effectively.

Web Server	Benefits	Limitations
Amazon Web Services (AWS)	• Offers a comprehensive suite of cloud services • Has large network of data centers • Most mature and has the largest market share	• Complex to navigate and manage with its vast array of services • Expenses can quickly escalate if not properly managed
Microsoft Azure	• Easy integration with Microsoft products	• Specifically designed for business customers
Google Cloud Platform	• Strong offerings in AI and machine learning	• Has fewer global data centers

Table 9: Web Server Decision Table

4.8. Web Application

Web application is an important part in our project, since it provides the user interface that will be used to display our data and predictions to end users. Our team decides on using React as the foundation to build upon our application. The React library provides vast reusable components and code abstraction for building user interfaces, and is widely adapted by developers. React has a smooth learning curve and our team has previous experience with using it, which enables us to efficiently build a functional application. In addition, we consider the scalability as our client required, and for the software part, it should be able to address the growth in size and complexity. React relies on third-party tools but developers can easily implement a maintainable architecture with server-side rendering in assist of scaling the application.

Alternatives to React are Angular and Vue. Angular has been a mature framework with good contributors and complete packages for application development. However, it has a steeper learning curve due to its comprehensive feature set and TypeScript base that is not feasible given our time constraint. Although Vue is known for its simplicity and detailed documentation, making it highly accessible and easy to grasp for newcomers, it uses template-based syntax and these templates are less supportive to reusability in large applications. Considering the possibility of scaling our project in the future, our team selected React.

Web Application	Ease of Implementation	Developer's Community	Scalability	Chosen?
Angular	Low	Large	High	No
React	High	Very Large	High	Yes
Vue	High	Small	Low	No

Table 10: Web Application Decision Table

4.9. Machine Learning Algorithm

A Machine Learning Model will be used in our project to determine the presence of a forest fire. The ML model will take the near real-time data gathered from our IoT sensor as an input, and predict the likelihood of a forest fire in the region. The main decision here is to choose a machine learning algorithm. Our group considers Linear Regression [22], Random Forest [23], Support Vector Machine [24], and Deep Learning [25] as our options, and analyzed their computational efficiency, accuracy, and implementation difficulty, and finally decides on implementing a Random Forest model, since it allows us to handle non-linear relationships, is computational efficient in both training and testing which is good for real-time applications, and it is relatively easy to implement in comparison with other algorithms given our time constraints. Random forests are also a classification algorithm which is especially applicable for our design as it needs to classify whether the data readings indicate a fire or not.

Machine Learning Model	Efficiency	Accuracy	Implementation Difficulty	Chosen?
Linear Regression	Computationally efficient in both training and prediction	Only good for linear relationships	Easy	No
Random Forest	Computationally efficient in both training and prediction	Capable of handling non-linear relationships, less likely to overfit	Easy	Yes
Support Vector Machine	Computationally intensive in training, but	Effective in high-dimensional spaces	Moderate	No

	efficient in prediction			
Deep Learning	Computationally intensive in both training and prediction	High	Hard	No

Table 11: Machine Learning Model Decision Table

5. Implementation

5.1. Implementation Overview

The implementation plan of our solution is outlined in figure 14 below. As previously mentioned the colours of figure 14 correspond with the functional block diagram. This design involves the deployment of the physical device (indicated by the lighter green boxes), consisting of the IoT sensors and the microcontroller, in a remote location where the user desires the detection of forest fires. Within 10km of this device, a LoRaWAN gateway is set up indoors (most likely inside the user's house or facility) where it can be easily accessed. The IoT device will transmit the temperature, humidity, and gas data it collects to the gateway via the microcontroller's LoRa radio. The gateway will be connected to wifi as the data will be put onto The Things Network. The AWS web server will pull this information and store it in a database. This data will be displayed on our web application on a map and on graphs where the user can easily view the information (per functional requirement 4.1.1). The web server will also make API calls to OpenWeather which will provide us with an FWI rating that can be displayed on the web application as an indication of fire risk.

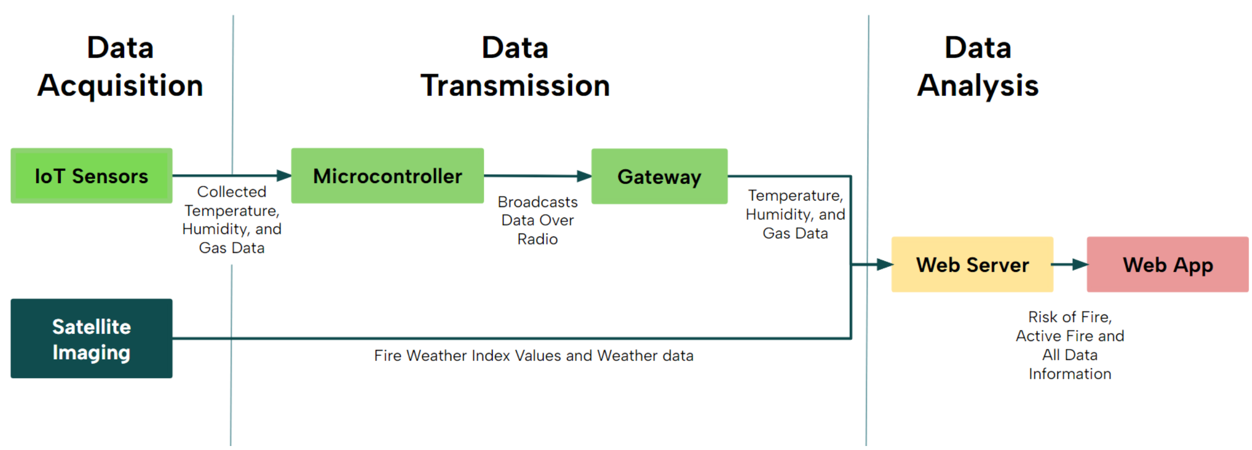


Figure 13: Implementation Overview

5.2. Communication Specifics

The communication specifics are outlined in figure 15 below. The sensor data collected will send bits of data over I2C to the Wio-E5 microcontroller. The data is converted into a protocol buffer with ten fields (one for each of the two BME688s readings of: temperature, humidity, pressure, CO2 ppm, and VOC ppm) [26]. This protocol buffer is transmitted as a HEX string and sent to the gateway using the MCU's LoRa radio. This data is secure as "LoRaWAN enforces using AES 128-bit message integrity checks and payload encryption" (per non functional requirement 4.2.5) [27]. This HEX data is sent to The Things Network through the gateway's WiFi connection where it can be encrypted, transmitted to the AWS server, and decrypted in base 64 via an MQTT payload. This communication is secure as it also utilizes The Things Stack payload encryption [22].

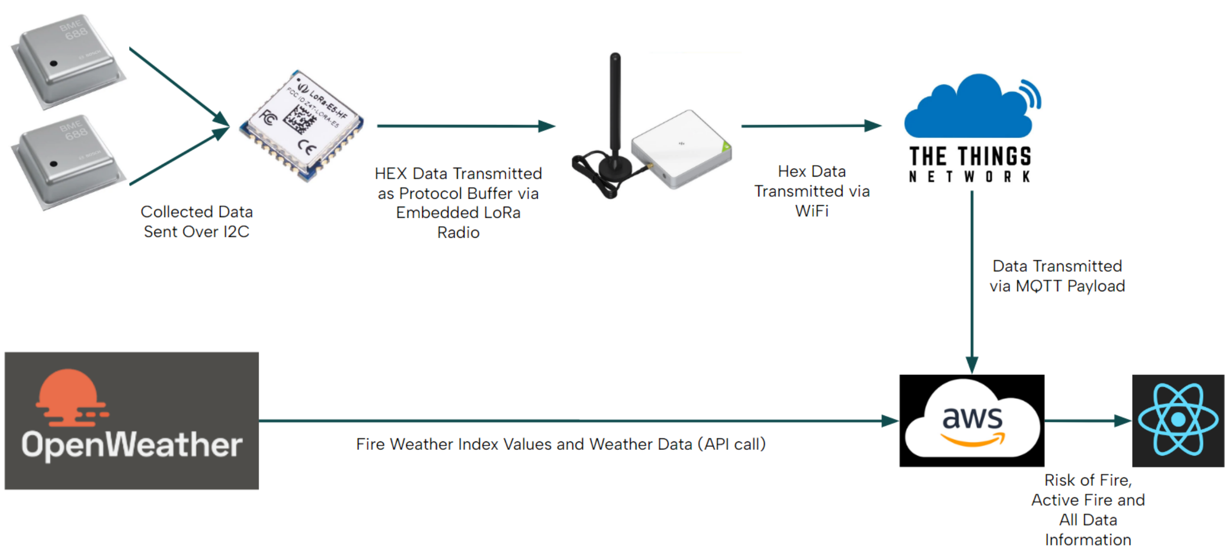


Figure 14: Communications Specifics

5.3. PCB

The team designed the PCB seen in figure 13. The dimensions of the PCB are designed to fit into the enclosure's PCB mount standoffs. The is a micro-USB connection, as seen on the left side of the PCB, which is used for both charging the batteries as well as supplying the board with power during early development stages of the board before a battery is connected. The JTAG connector is for direct connection with an ST-Link. The ST-Link is used for debugging and programming the STM32 microcontroller. An antenna cable connection can be seen at the top of the PCB. This allows for the implementation of an antenna cable with a waterproof gasket so that the antenna can be attached outside of the enclosure without sacrificing the enclosure's waterproofing. Although it is not used in the final product, the team decided to keep in a connection for the SHT-31 sensor in case unpredicted complications occurred with the BME688 sensor. The schematic for the PCB can be seen in Appendix A.6.

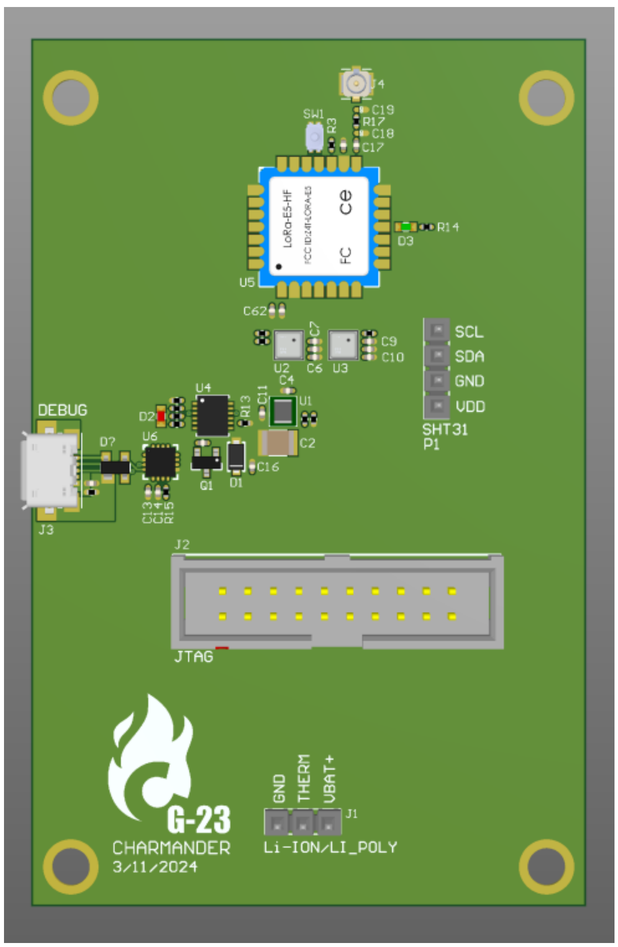


Figure 13: PCB

Two BME688 sensors were implemented in the final PCB design to ensure sensor redundancy in case one sensor fails (per functional requirement 4.1.4). They can be identified as the two gray squares under the microcontroller. This is accomplished by comparing the two sensor readings for each data point (temperature, humidity, pressure, CO2 ppm, and VOC ppm). The sensor with the reading more indicative of a fire will be used. This is because in our case false positive results are better than false negative results.

5.4. Software Implementation

The software implementation is outlined in Figure 16 below. The web application will utilize React for the user interface, and data visualization. We will also use Leaflet.js which is an open source library to create interactive maps using data from OpenStreetMap. This will be used to present sensor locations. AWS Amplify will be used for the deployment of the application. As for the web server, it will contain Amazon DynamoDB. This database program has been chosen due to its integration with other AWS services. The database stores all sensor data from The Things Network and satellite data from OpenWeather API. The web server will also have AWS lambda which acts as the backend of our web application. We have deployed a pre-trained machine learning model on Amazon S3, designed to assess fire risk levels. Upon receiving new sensor data, this model processes the information and automatically dispatches notifications to users identified at high risk of fire. This is responsible for collecting and processing the incoming data. Finally, we have API Gateway which acts as the bridge between the frontend web application and lambda functions. It allows the frontend to make REST API calls to endpoints and the API gateway maps the endpoint to the appropriate lambda function. Looking at the Open Weather API, it is able to provide Fire Weather Index (FWI), temperature, humidity, and wind speed and direction for a specified coordinate.

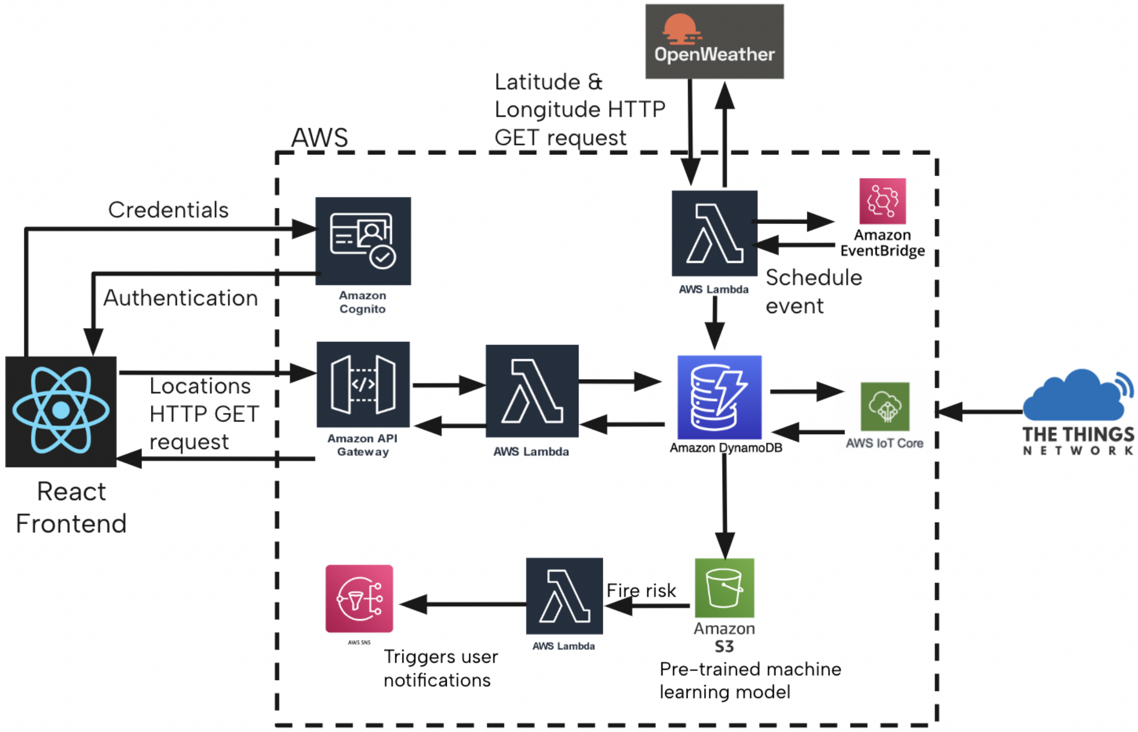


Figure 16: Software Implementation

5.5 Machine Learning Algorithm

As previously discussed, the team opted to use random forests as the machine learning algorithm. The algorithm was trained on a fire detection dataset with over 62,000 datasets [28]. This data set in particular was chosen as the collection of training data was performed across a large range of environments including: normal indoor, normal outdoor, indoor wood fire, indoor gas fire, outdoor wood/coal/gas grill, outdoor high humidity, and many more. This data set included features such as raw H2 and raw ethanol which our design did not have access to. However, the group found that using this training set the algorithm was still able to detect a fire with 99% accuracy utilizing the 5 features supplied by our IoT sensors (temperature, humidity, pressure, CO2 ppm, and VOC ppm).

The results of training our algorithm can be seen in the correlation matrix in figure 17. It can be seen that humidity and pressure have a strong positive relationship and VOC level and pressure have a strong inverse relationship. Overall, the two variables that had the most correlation to a fire being present were humidity and VOC level as indicated by the fire alarm column.

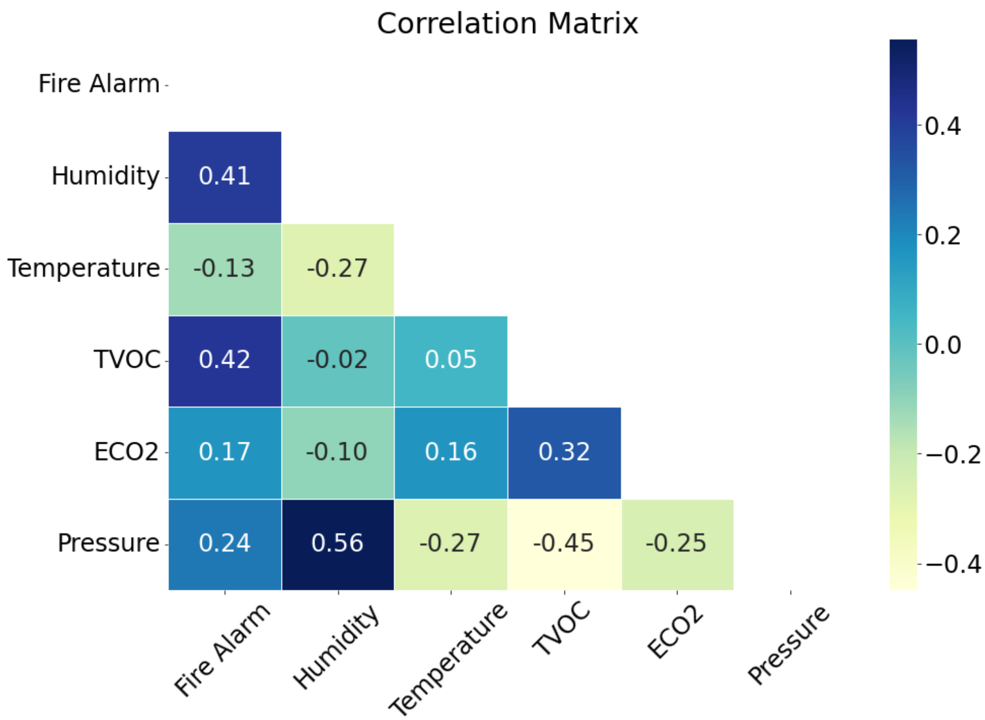


Figure 17: Correlation Matrix

5.6. Fire Detection and Prediction

The detection and prediction of forest fires is handled separately as the detection algorithm uses the near real time IoT sensor data and the prediction comes from satellite data.

5.6.1. Detection

A fire can be detected as our machine learning takes the IoT sensor data and classifies it as alarming or not. The implementation and test results of this machine learning model is further outlined in section 5.5. Should a fire be detected, the users will be immediately notified.

5.6.2 Prediction

A fire can be predicted by the Fire Weather Index System. By making an API call to OpenWeather, an FWI rating value will be returned. This will be a numerical rating indicating the risk of a fire occurring in the forested area.

Appendix A

A.1. Raspberry Pi

Raspberry Pi 5

https://www.raspberrypi.com/documentation/computers/raspberry-pi-5.html

A.2. SenseCAP

SenseCAP Wireless Air Temperature and Humidity Sensor – LoRaWAN

https://solution.seeedstudio.com/product/sensecap-lorawan-air-temperature-and-humidity-sensor/

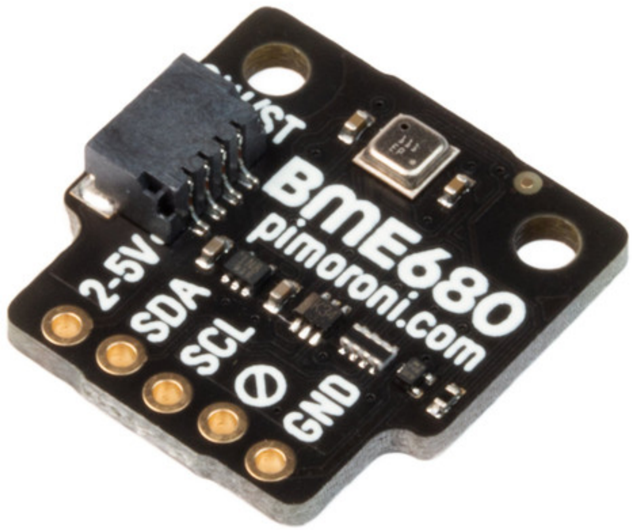
A.3. DFRobot's SEN0385 SHT-31

https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/5337/SEN0385_Web.pdf

A.4. Amphenol Telaire T9602-3-D

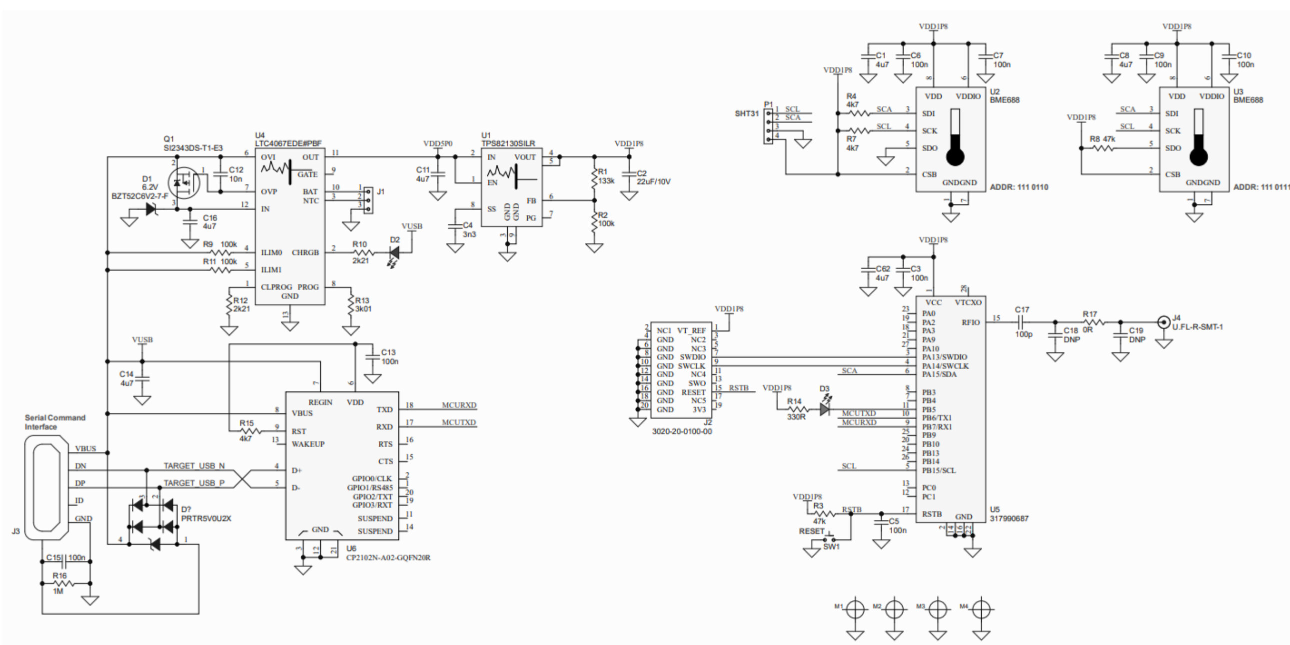
https://www.amphenol-sensors.com/hubfs/Documents/AAS-920-638F-Telaire-T9602-060316-web.pdf

A.5. BME680 Eval Board Sensor



https://cdn.shopify.com/s/files/1/0174/1800/files/bst-bme680-ds001.pdf?v=1629474140

A.6. PCB Schematic



A.7. Waterproof Antenna Cable Connector

https://www.seeedstudio.com/UF-L-SMA-K-1-13-120mm-p-5046.html

A.8. Waterproof Vent Plug

https://www.polycase.com/ua-006

References

[1] “Home Page: Community Health Cloud Innovation Centre,” Cloud Innovation Centre, https://cic.ubc.ca/

[2] British Columbia, “Wildfire Averages - Province of British Columbia,” Gov.bc.ca, 2017. https://www2.gov.bc.ca/gov/content/safety/wildfire-status/about-bcws/wildfire-statistics/wildfire-averages

[3] The Things Network, “What are LoRa and LoRaWAN?,” *The Things Network*. https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/

[4] “Research applications laboratory,” Improving Wildfire Prediction with Cutting-Edge Satellite Imagery and Artificial Intelligence (AI) | Research Applications Laboratory, https://ral.ucar.edu/solutions/benefits/improving-wildfire-prediction-cutting-edge-satellite-imagery-and-ai

[5] N. R. Canada, “Canadian wildland fire information system: Data sources and methods for daily maps,” Canadian Wildland Fire Information System | Data Sources and Methods for Daily Maps, https://cwfis.cfs.nrcan.gc.ca/background/dsm/fm3

[6] Natural Resources Canada, “Canadian Wildland Fire Information System,” *Nrcan.gc.ca*, 2020. https://cwfis.cfs.nrcan.gc.ca/home

[7] N. R. Canada, “Canadian wildland fire information system: Canadian forest fire weather index (FWI) system,” Canadian Wildland Fire Information System | Canadian Forest Fire Weather Index (FWI) System, https://cwfis.cfs.nrcan.gc.ca/background/summary/fwi

[8] B. W. Service, “Wildfire Glossary,” Province of British Columbia, https://www2.gov.bc.ca/gov/content/safety/wildfire-status/about-bcws/glossary#fuel

[9] “AN-02P Diecast Aluminum NEMA Enclosure,” https://www.polycase.com/an-02p

[10] “Propagation loss,” Propagation Loss - an overview | ScienceDirect Topics, https://www.sciencedirect.com/topics/computer-science/propagation-loss

[11] SenseCAP M2 Multi-Platform Gateway Overview | Seeed Studio Wiki," wiki.seeedstudio.com, Aug. 28, 2023. https://wiki.seeedstudio.com/Network/SenseCAP_Network/SenseCAP_M2_Multi_Platform/SenseCAP_M2_Multi_Platform_Overview/

[12] The Things Network. "The Things Network." *Thethingsnetwork.org*, The Things Network, 2019, www.thethingsnetwork.org

[13] "LoRaWAN-network-architecture," *www.researchgate.net*. https://www.researchgate.net/figure/LoRaWAN-network-architecture_fig1_341298152

[14] "SenseCAP Wireless Air Temperature and Humidity Sensor - LoRaWAN," *Seeed Studio IIoT Solutions*. https://solution.seeedstudio.com/product/sensecap-lorawan-air-temperature-and-humidity-sensor/

[15] "Wio-E5 mini Dev Board - STM32WLE5JC, ARM Cortex-M4 and SX126x embedded, supports LoRaWAN on EU868 & US915," www.seeedstudio.com, Jan. 26, 2024. https://www.seeedstudio.com/LoRa-E5-mini-STM32WLE5JC-p-4869.html?queryID=1aeb7a99a575e44a6a0c3e46fa8bbf35&objectID=4869&indexName=bazaar_retailer_products

[16] "LoRa-E5 LoRa Wireless Module -Powered by STM32WE5 Datasheet V1.0." Available: https://files.seeedstudio.com/products/317990687/res/LoRa-E5%20module%20datasheet_V1.0.pdf

[17] B. W. Service, "Fire weather index system," Province of British Columbia, https://www2.gov.bc.ca/gov/content/safety/wildfire-status/prepare/weather-fire-danger/fire-weather/fire-weather-index-system

[18] "Bosch Sensortec | BME688 Datasheet 1 | 60." Available: https://cdn.shopify.com/s/files/1/0174/1800/files/bst-bme688-ds000.pdf?v=1620834794

[19] "The BSEC Software," Bosch Sensortec. https://www.bosch-sensortec.com/software-tools/software/bme680-software-bsec/

[20] "Understanding volatile organic compound emissions from wildfires in the western United States | NOAA Climate.gov," www.climate.gov. https://www.climate.gov/news-features/feed/understanding-volatile-organic-compound-emissions-wildfires-western-united#:~:text=Volatile%20organic%20compounds%20(VOCs)%20are

[21] OpenWeatherMap.org, "Current weather and forecast - OpenWeatherMap," Openweathermap.org, 2012. https://openweathermap.org/

[22] IBM, "About Linear Regression | IBM," www.ibm.com, 2023. https://www.ibm.com/topics/linear-regression

[23] BM, "What is Random Forest? | IBM," www.ibm.com, 2023. https://www.ibm.com/topics/random-forest

[24] "What is support vector machine? | IBM," www.ibm.com. https://www.ibm.com/topics/support-vector-machine

[25] IBM, "What is Deep Learning?," www.ibm.com. https://www.ibm.com/topics/deep-learning

[26] "Protocol Buffers," Protocol Buffers Documentation. https://protobuf.dev/

[27] "Network Security," The Things Network. https://www.thethingsnetwork.org/docs/network/security/

[28] "Smoke Detection Dataset," www.kaggle.com. https://www.kaggle.com/datasets/deepcontractor/smoke-detection-dataset