Satellite Fire Patrol

Technological Feasibility
September 21, 2023

Team Members:
Aaron Santiago, Ricardo Chairez, and Zachary Hallemeyer

Clients:
Benjamin Wiebe, Dr. Camille Gaillard, and Dr. Christopher Doughty

Mentor:
Saisri Muttineni

Table of Contents

Introduction

The Hawaiian islands hold immense ecological and cultural value but are currently threatened by increasing hot and dry conditions caused by climate change. This threat was evident in August 2023 when multiple wildfires across Maui resulted in at least 106 deaths, billions of dollars in damages, and ecological destruction. These fires begin in grasslands, which are vulnerable to high temperatures due to increasing drought and low moisture. Other Hawaiian ecosystems such as coral reefs and tropical forest canopies are also threatened by rising temperatures resulting from climate change.

With climate change being such a major threat to the Hawaiian islands, we look to high-resolution thermal remote sensing as a way to monitor ecological threats. It has the potential of alerting researchers and natural resource managers when temperatures are approaching dangerous levels, so they can help mitigate potential damages. A thermal infrared image of Maui from NASA's Landsat sensor on August 7th, 2023 showed grassland temperatures ranking in the top 4% of the last decade. The information to detect a potential ecological disaster was there, but failed to reach the people it needed to.

Issues with Current System

- Lack of warning system for areas sensitive to excessive heat.
- Ecosystems are dying due to high temperatures
- No mitigation system for major loss of life or infrastructure

Team Satellite Fire Patrol is partnering with clients Dr. Christopher Doughty, Dr. Camille Gaillard, and PhD student Benjamin Wiebe to create a GUI web application that uses real-time satellite thermal data to identify warning signs in the Hawaiian islands and send alerts to the proper authorities. The app will automatically aggregate and process satellite thermal data from multiple sources, compare historical averages to highlight temperature anomalies, and present the data in a user-friendly interface for a range of resource management applications.

Our proposed Solution

- Using NASA data to warn and highlight abnormal temperatures
- Prevent fires by determining which areas have excessive thermal activity
- Make the data as available as possible to prevent as many fires as possible

In our development, we must consider the technological challenges we face and what technologies we will use to solve them. In this document, we will identify all the technical challenges we face, analyze each challenge, provide a chosen approach, look at alternative solutions, and prove feasibility by stating how we intend on implementing said solution. In the technology integration section, we will explain how all the chosen technologies work with one another. The goal of this document is to select the different technologies we will be using and look at how they all work together.

Technological Challenges

Introduction
In this section, we will introduce the main requirements of the project. This will be accomplished by listing the challenges we anticipate, and then we will further explore these challenges.

Technical Requirements
1. Store and retrieve historical thermal data
2. Receive, process, and store near real-time thermal data from NASA satellites
3. User authentication
4. Real-time custom alerts to users
5. Overlaying thermal data

The main features of our web application rely on a working solution to these requirements. For this reason, these requirements must be explored, and a high level plan must be established.

a. Historical Data
Our clients have expressed the interest in storing and using historical thermal data in the web application. Mainly, the clients want to use this data to filter out inaccurate readings from the real-time thermal data. Secondly, as a stretch goal, the clients want to display the historical data on the web app for users to review how thermal data changes over time. For these reasons, historical data will have to be stored in some relation to the web application.

b. Real-time Data
Near real-time data from NASA satellites will have to be delivered and processed by the web application in a reasonable amount of time. The data is officially located at LPDAAC (a NASA data server) available through the APPEARS API.

c. User Authentication
The web application will be able to be used by anyone. However, certain features such as custom alerts will require private user accounts. The challenge here is to provide secure user authentication and storing said user data.

d. Custom Alerts
Custom alerts for when a temperature reading for a custom region exceeds a user given threshold. Because of this, the web application will have to check new temperature readings with the custom thresholds. The web application will also have to alert the user via email or SMS messaging, with phone alerts being a stretch goal.

e. Overlaying Thermal Data
The web application main feature is overlaying thermal data from NASA satellites onto a virtual map. For this reason, the thermal data will need to be processed into a Cloud optimized geotiffs (COGs) that can be displayed on top of a map API.

Conclusion

Our project will be using years of historical data in addition to near real time NASA data to display risk areas on an easy-to-use user interface. Users will be able to log in to save multiple areas to monitor on the web application. Users will also be able to view and set areas to monitor that will alert them if the area they selected is in danger. Now that the main challenges from the desired features have been addressed, we may now begin to explore the possible solutions to these challenges. In the next section, we will discuss the path to implementation.

Technological Analysis

Historical Data

Introduction

In this section, we will explore the challenges and possible solutions that were explored in the previous section. We will accomplish this by analyzing the alternatives and deciding which approach we should take to address our requirements.

Store and Retrieve Historical Data

1. The Issue

   We will need a way to filter historical data into a simplified form and store said data. It will need to be available both for displaying on the front end and for use in quality control in the backend.

2. Alternatives

   With data storage being a complex issue, there is a large amount of methods that can be used to solve this problem. For instance, the historical data can be stored in a third party database such as Anaconda and AWS.

3. Chosen Approach

   In order to solve this problem, we need to remember we will be using NAU monsoon for data processing. As NAU monsoon allows for large amounts of data to be stored and accessed. With the ability to use python in NAU monsoon, the data will be able to be processed efficiently with NumPy, GDAL, rasterio and pandas. For the reasons above, we will be able to store the historical data on NAU monsoon using python and local storage. This allows both the front end and back end of the web application to request and use this data.

4. Proving Feasibility

   In order to test the feasibility of this approach, a test database can be implemented into NAU monsoon. This test database will be attempted to be accessed and store new information. If the test is successful, then NAU monsoon will be used for data filtering, storing, and accessing.

Near real-time Satellite Data

1. The Issue

   We will need to access NASA Ecostress data in a near real-time manner. The data will need to be processed, stored, and sent to the front end to refresh the web application's display.

2. Alternatives

   - **AWS:** Alternative Approaches are managing the data using Amazon's AWS and hosting our website all on the same data server. This would require a cost for hosting and using the required amount of data.
   - **Anaconda:** Python and R management for data good for data management does not include hosting.

3. Chosen Approach

   In order to store the data, we will need to use a cost-effective and large database. Moreover, the clients want our data processing to be executed in python. For the backend we will be using Django which is a Python framework that will be hosted on NAU's Monsoon which is capable of processing data and holding large datasets. For the frontend, we will be using Vue.js for a flexible and scalable display of the data. With our maps and mapping, we will be taking advantage of Mapbox's free global API to display our data and allow users to select portions of the map or surveillance.

4. Proving Feasibility

   In order to test the feasibility of this approach, a test database can be implemented into NAU monsoon. This test database will be attempted to be accessed and store new information. If the test is successful, then NAU monsoon will be used for data filtering, storing, and accessing.

User Authentication

1. The Issue

   Users will need to be able to create accounts and have the system store relevant data about their preferences. The solution must be secure, so the likelihood of leaking personal information is extremely low.

2. Alternatives

   - **DUO Mobile:** Using Duo mobile would allow for two-factor authentication and is something that all members of the university have set up. Using DUO would give everyone in the school a relatively easy transition to using the web application and saving their results.

   - **Facebook Authentication:** Facebook authentication would also be a popular approach since Facebook is a popular social media platform that most people have access to. Using Facebook would allow users to have a secure login without worrying about us keeping or using their passwords.

Scale (1 - 5) worst to best

	Price	Security	Limits
Google Accounts	5	5	5
DUO Mobile	2	5	1
Facebook Accounts	2	5	2

3. Chosen Approach

The chosen approach will be to use Google account API. This API allows all user credentials to be stored in Google rather than in our own servers in NAU monsoon. This will make the user accounts inherently secure. Moreover, we will be able to user these accounts to keep preferences anonymous.

4. Proving Feasibility

In order to test the feasibility of this approach, test accounts will be created in order to store account preferences onto NAU monsoon. We will create a fake website that has a login to test out the Google API approach and display a success message once the user has logged in using Google.

Real-time Alerts

1. The Issue

The issue is the clients specified they wanted notifications whenever a certain region exceeds a certain temperature. To accomplish this, they wanted SMS and email notifications available for users.

2. Alternatives
    - **WebPushr:** Free API service that only sends notification within the web browser. It is free up to a certain amount of notifications and reliable.
    - **PushEngage:** Highly rated and free for up to 200 subscribers and 30 lifetime notifications. Has custom messages and notifications allows SMS and Email notifications for a subscription.

Scale (1 - 5) worst to best

	Price	Notifications Ability	Limits
WebPushr	3	1	1
PushEngage	2	5	2
OneSignal	5	5	3

3. Chosen Approach

    We wanted an approach that was free and included SMS and email notifications. We selected OneSignal as our API because it has unlimited SMS notifications and up to 5,000 free email notifications, which we are hoping to not exceed while working on the capstone project.

4. Proving Feasibility

    To prove the feasibility of our chosen approach, we will create a push notification using SMS to test the OneSignal API. Then we will set it, so it only pushes notifications once a user has selected a region to monitor for a certain temperature.

Overlaying Thermal Data

1. The Issue

    The historical and near real-time thermal data will be displayed on a map.

2. Alternatives

    Google Maps API would be suitable for this task as it is continually updated and documentation is readily available. However, this approach will be costly.

Scale (1 - 5) worst to best

	Price	Detail	Customization
Google Maps API	2	5	2
Mapbox API	5	5	5
Bing Maps API	2	5	2

3. Chosen Approach

    An open source map API called Mapbox is both free and has a good deal of documentation. Using this API will be more cost-effective and the API has been continually updated and improved with it being open source.

4. Proving Feasibility

    In order to test feasibility, actual thermal data from NASA Ecostress will be displayed as a layer on top of the Mapbox. If this is accomplished, the API can be used for the project.

Conclusion

For our technological analysis, we covered storing and analyzing historical data that will be used to display information about temperatures using MapBox. We will highlight risk zones and allow users to monitor custom zones with One Signal. We will be using Google Accounts to Authenticate and user logins. Now that the potential solutions to our technical challenges have been discussed, we may now explore how these solutions will interact with each other. In the next section, the system as a whole will be discussed.

Technological Integration

Introduction

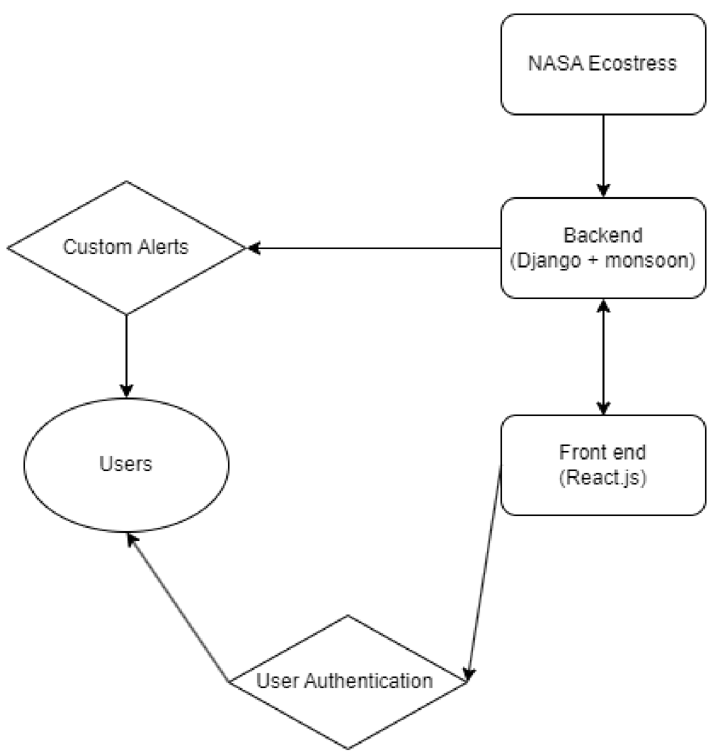
Our team will be managing large datasets and large amounts of incoming data. To properly manage all of it, we will be using multiple technologies that need to work together to create a system that manages and displays our data in an easy-to-use and understandable web application. Web applications require lots of running parts and software programs to work in congruence to create an optimal system that addresses the problem at hand.

How it all comes together

Our solution will involve collecting and displaying lots of data in the form of a web application. We will be accepting approximately 10 GB of data per week from NASA's satellite and displaying relevant temperature readings on an intuitive user interface online. We will be having the data collected and organized on NAU's Monsoon, from there we will form a bridge between Monsoon's data and our web application that allows users to request areas to be surveyed. Django will be used for the backend, since it works well with Python and is scalable, which is something that the clients requested for the future. For our web application, we will be using Vue. Js because of its flexibility and how it works well with Django as well.

Using approximately 5 years of environmental data, we will organize and create averages to establish the basis of our alert system. If temperatures exceed what is normal then we will alert users with OneSignal which is a free notification system, our preferred method of notifying users will be using SMS, with email as a secondary form of communication.

System Diagram



Conclusion

With the increase in wildfires and rising coral reef temperatures, the Hawaiian islands are vulnerable to the threat of climate change. Team Satellite Fire Patrol is working on a project with clients Dr. Camille Gaillard, PhD student Benjamin Wiebe, and Dr. Christopher Doughty to create a software that helps identify warning signs in the Hawaii area from real-time satellite thermal data.

Our software must address the following technical challenges:

- Historical data
- Real time data
- User authentication
- Custom alerts
- Overlaying thermal data

This document outlines the analysis and research we put into finding solutions for each of these challenges. We aimed to pick solutions that were scalable, reliable, and available to us. We believe we chose the best technologies that are available to us.

- NAU's Monsoon
- Django
- Google Account API
- OneSignal
- Mapbox

In conclusion, our proposed solution will involve multiple open-source technologies that will take advantage of NASA's data to create an early warning system for wildfires to allow for the mitigation of ecosystem loss and wildfire damages.