SIGNAL SAFARI

PROJECT REPORT

Submitted By

AKSHITH S – 2022107014 - GI

CHRIST SAMUEL H – 2021107004 - GI

HUSSWATH L S U – 2021105515 - ECE

KRITHIKA V – 2022107016 - GI

MADESHWAR R – 2021105524 - ECE

PREETHIKA M – 2021105036- ECE

SHAKTHI SRIRAM S – 2021107022 - GI

SRI DHARANI R – 2022115074 - IT

SWATTIK MUKHOPADHYAY – 2022107036 - GI

CEG Tech Forum

An ISO 9001:2015 certified organisation

CEG TECH FORUM

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2024

BONAFIDE CERTIFICATE

Certified that this project report **SIGNAL SAFARI** is the bonafide work of **AKSHITH S, CHRIST SAMUEL H, HUSSWATH L S U, KRITHIKA V, MADESHWAR R, PREETHIKA M, SHAKTHI SRIRAM S, SRI DHARANI R, SWATTIK MUKHOPADHYAY** who carried the project work under my supervision.

Sauman Raaj S	Sahai Jordi Alan A	Keerthana B K
Student Director	Student Director	Student Director

Dr. K. Srinivasa Raju	Dr. K. Rathnakannan	Dr. K. Vani
Project Guide,	Staff Coordinator	Staff Coordinator
Institute of Remote Sensing	Professor	Professor
Department of Civil Engineering	Department of EEE	Department of IST
College of Engineering, Guindy	College of Engineering, Guindy	College of Engineering, Guindy
Anna University, Chennai – 600 025	Anna University Chennai – 600025	Anna University Chennai – 600025

ANNA UNIVERSITY

CHENNAI 600 025

Degree	:	Bachelor of Engineering (B.E)
Organization	:	CEG TECH FORUM
Name of the project	:	SIGNAL SAFARI
Name of the Students	:	AKSHITH S CHRIST SAMUEL H HUSSWATH L S U KRITHIKA V MADESHWAR R PREETHIKA M SHAKTHI SRIRAM S SRI DHARANI R SWATTIK MUKHOPADHYAY
Name and Designation of Guide	:	Dr. K. SRINIVASA RAJU Professor, Institute of Remote Sensing Department of Civil Engineering, College of Engineering, Guindy Anna University, Chennai – 600 025

ACKNOWLEDGEMENT

We are highly indebted to our guides, **Dr. K. Srinivasa Raju,** Professor, Institute of Remote Sensing, Department of Civil Engineering, College of Engineering, Guindy, Anna University for his valuable guidance, insightful comments, support, encouragement and keen interest which helped us in the successful completion of this project.

We want to express our sincere gratitude to CEG TECH FORUM, College of Engineering Guindy, for your instrumental role in administering and overseeing the project.

Finally, we would like to express our heartfelt gratitude to our Family and Friends for their prayers, encouragement, support and their consistent belief in us.

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

Signal Safari embarks on an extensive expedition to thoroughly assess signal integrity from the primary cellular provider, Airtel, across the expansive terrain of the CEG campus. Our ambitious project centres on developing a dynamic, automated mapping framework to intricately chart the ever-changing network coverage in real-time. Leveraging state-of-the-art technology, we aim to offer a visually immersive experience that highlights subtle variations in signal strength, fostering a deep understanding of network performance for the diverse community of students and administrators navigating the campus landscape. Through this initiative, we strive to equip stakeholders with actionable insights, facilitating informed decision-making and enriching the overall connectivity experience within the campus ecosystem.

INTRODUCTION

Signal Safari embarks on a comprehensive expedition to meticulously gauge the signal integrity emanating from the principal cellular operator, Airtel, spanning the vast terrain of the CEG campus. Our ambitious project is anchored on the creation of a dynamic, automated cartographic framework that intricately maps out the ever-fluctuating network coverage in real-time. By harnessing cutting-edge technology, we aspire to provide a visually immersive experience that illuminates the nuanced variations in signal strength, thus fostering a profound understanding of network performance for the diverse community of students and administrators traversing the campus landscape. Through this endeavour, we endeavour to empower stakeholders with actionable insights, enabling informed decision-making and enhancing the overall connectivity experience within the campus ecosystem.

NEED FOR THE PROJECT

Poor mobile the leading problems among college students. Mapping mobile networks across college campuses is essential for ensuring seamless connectivity and improving communication infrastructure. The need for such a project stems from the increasing reliance on mobile devices for academic, social, and professional activities. By comprehensively mapping out the network coverage, signal strengths, and potential dead zones, universities can identify areas for improvement and optimize their network infrastructure. This project aims to enhance the overall student experience by providing reliable connectivity for accessing educational resources, communicating with peers and faculty, and participating in campus events. Additionally, understanding mobile network dynamics can facilitate emergency response efforts and enable campus administrators to make informed decisions regarding network upgrades and expansions. Ultimately, this initiative serves to create a more connected and efficient campus environment conducive to learning and collaboration.

CHAPTER 1
PHASE I

1.1 INTRODUCTION

The main objective of this project is to determine the network coverage of cellular operators across the campus. It is not necessary to measure the strength of signal at each and every point. As we know, the strength of signal varies dynamically and is unpredictable even within a short interval of time. So, data collection at each and every point is not a feasible option and is found to be a time-consuming and tiring process. During these types of situations, the method of interpolation is very useful in determining the values of unknown data points using the values of a few known data points.

Initially, a few locations at which the signal strength is drastically varying and points having a constant strength are selected at equal numbers. The points of interest are chosen in such a way that the points are more or less equally spread across the college campus. We chose 9 points after considering the above constraints.

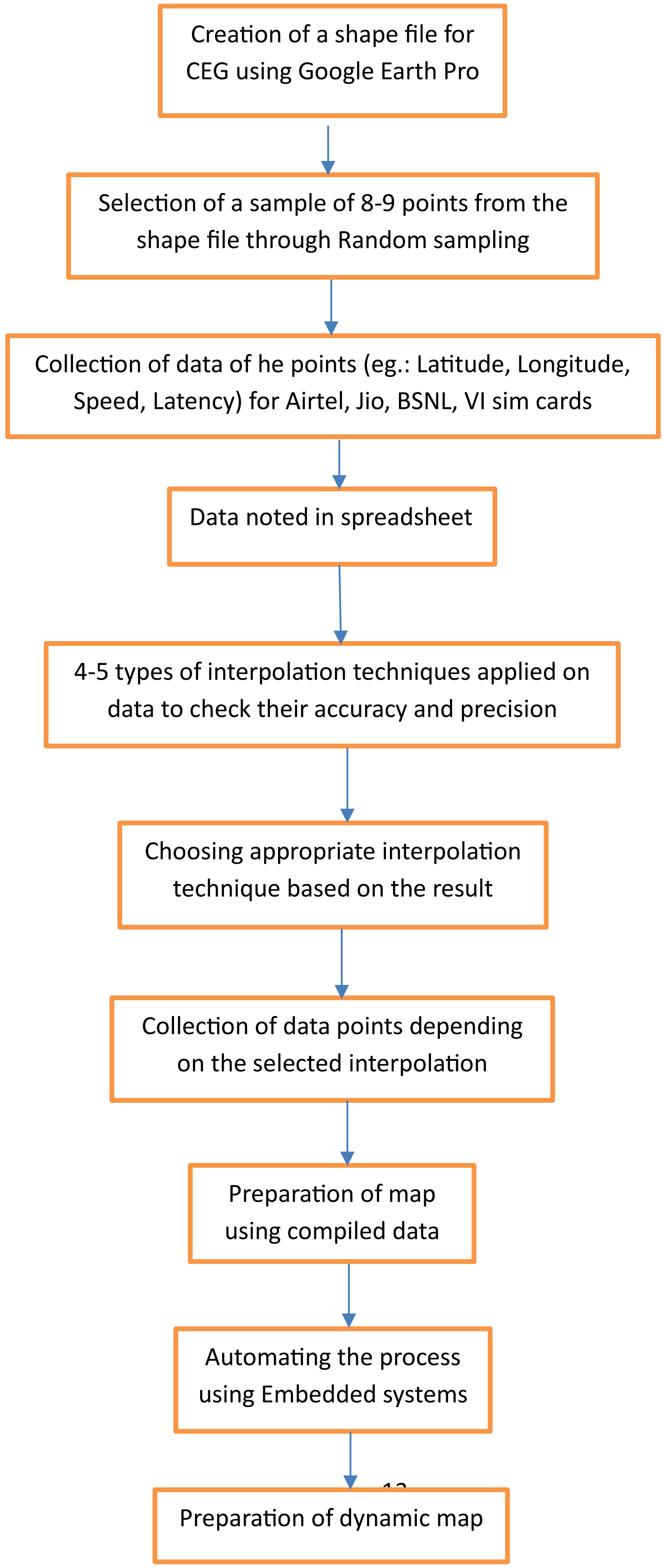
The points of interest were:

- Aambal Hostel
- Centre for Environment Studies (CFES)
- Hostel Office
- CEG Canteen
- Science and Humanities Building
- J Block
- Centre for Distance Education (CFDE)
- Anna Statue
- Anna University Sports Board (AUSB)

A Data Collection Survey was planned and executed on the evening of 21st December, 2023. Prior to the data collection, volunteers and friends who were interested to help for collecting data were carefully selected in such a way that the group of volunteers had at least one person using one of the four major cellular operators: Jio, Airtel, VI, BSNL. Another factor that was considered in selecting the volunteers was ensuring that two persons from Jio and Airtel operators had compatibility to 4G and 5G spectrums.

The strength of the individual cellular operators was measured using OpenSignal, an open-source app. The reason for choosing OpenSignal over Ookla, a commonly used and popular open-source app to measure signal strength was because, OpenSignal had the feature of measuring the latency of the cellular operator apart from identifying the download and upload speed. Though Ookla has the feature of measuring latency, it was found to be not very much accurate compared to OpenSignal. The data collection process was conducted in such a way that only one volunteer of a particular cellular operator measured the signal strength. This was done to ensure that the strength of the signal does not get altered from the original due to voluntary disturbances from other volunteers. But it is to be noted, that the strength of the signal can be varying due to the hindrances of other users who were currently using the operator at that particular location at that instant of time.

1.2 METHODOLOGY



CHAPTER 2

INTERPOLATION TECHNIQUE

2.1 INTRODUCTION

Interpolation is a method of deriving a simple function from the given discrete data set such that the function passes through the provided data points. This helps to determine the data points in between the given data ones. This method is always needed to compute the value of a function for an intermediate value of the independent function. It is mostly used to predict the unknown values for any geographical related data points such as noise level, rainfall, elevation, and so on.

The unknown value on the data points can be found using the linear interpolation and Lagrange's interpolation formula.

The Linear interpolation formula is given by

$$y = y_1 + \frac{x - x_1}{x_2 - x_1} \times (y_2 - y_1)$$

Similarly, the Lagrange's interpolation formula is given as:

$$y = \frac{(x-x_1)(x-x_2).....(x-x_n)}{(x_0-x_1)(x_0-x_2).....(x_0-x_n)}y_0 + \frac{(x-x_0)(x-x_2).....(x-x_n)}{(x_1-x_0)(x_1-x_2).....(x_1-x_n)}y_1 + .. + \frac{(x-x_1)(x-x_1).....(x-x_{n-1})}{(x_n-x_0)(x_0-x_1).....(x_n-x_{n-1})}y_n$$

Fig 2.1 LAGRANGE'S INTERPOLATION FORMULA

2.2 INTERPOLATION METHODS

There are different types of interpolation methods. They are:

- **Linear Interpolation Method** – This method applies a distinct linear polynomial between each pair of data points for curves, or within the sets of three points for surfaces.

- **Nearest Neighbour Method** – This method inserts the value of an interpolated point to the value of the most adjacent data point. Therefore, this method does not produce any new data points.
- **Cubic Spline Interpolation Method** – This method fits a different cubic polynomial between each pair of data points for curves, or between sets of three points for surfaces.
- **Shape-Preservation Method** – This method is also known as Piecewise Cubic Hermite Interpolation (PCHIP). It preserves the monotonicity and the shape of the data. It is for curves only.
- **Thin-plate Spline Method** – This method consists of smooth surfaces that also extrapolate well. It is only for surfaces only

- **Biharmonic Interpolation Method** – This method is applied to the surfaces only.

- **Inverse Distance Weighted (IDW) Interpolation:**

Inverse Distance Weighting (IDW) interpolation estimates unknown values with specifying search distance, closest points, power setting & barriers. It explicitly makes the assumption that things that are close to one another are more alike than those that are farther apart. To predict a value for any unmeasured location, IDW uses the measured values surrounding the prediction location. The measured values closest to the prediction location have more influence on the predicted value than those farther away. IDW assumes that each measured point has a local influence that diminishes with distance.

$$z_p = \frac{\sum_{i=1}^{n} \left(\frac{z_i}{d_i^{\,p}} \right)}{\sum_{i=1}^{n} \left(\frac{1}{d_i^{\,p}} \right)}$$

Fig 2.2 IDW Interpolation Formula

2.3 THE POWER FUNCTION

As mentioned above, weights are proportional to the inverse of the distance (between the data point and the prediction location) raised to the power value p. As a result, as the distance increases, the weights decrease rapidly. The rate at which the weights decrease is dependent on the value of p. If $p = 0$, there is no decrease with distance, and because each weight λ_i is the same, the prediction will be the mean of all the data values in the search neighbourhood. As p increases, the weights for distant points decrease rapidly. If the p value is very high, only the immediate surrounding points will influence the prediction.

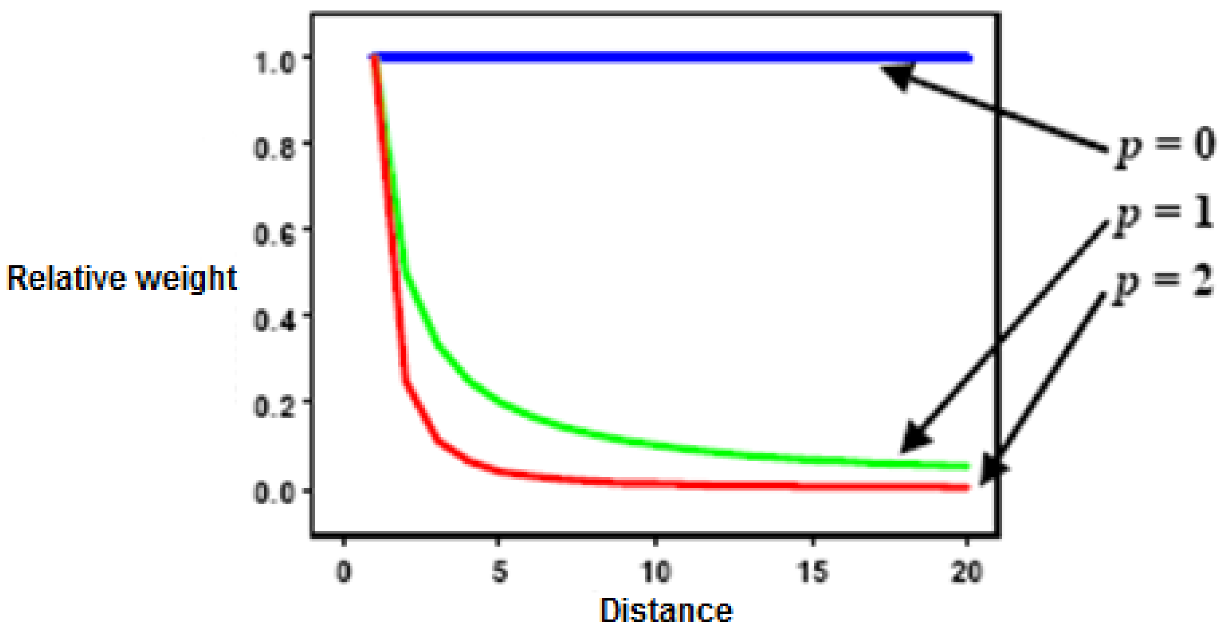


Fig 2.3 Decrease of weight with distance illustration

2.4 THE SEARCH NEIGHBOURHOOD

Because things that are close to one another are more alike than those that are farther away, as the locations get farther away, the measured values will have little relationship to the value of the prediction location. To speed calculations, you can exclude the more distant points that will have little influence on the prediction. As a result, it is common practice to limit the number of measured values by specifying a search neighbourhood. The shape of the neighbourhood restricts how far and where to look for the measured values to be used in the prediction. Other neighbourhood parameters restrict the locations that will be used within that shape. In the following image, five measured points (neighbours) will be used when predicting a value for the location without a measurement, the yellow point.

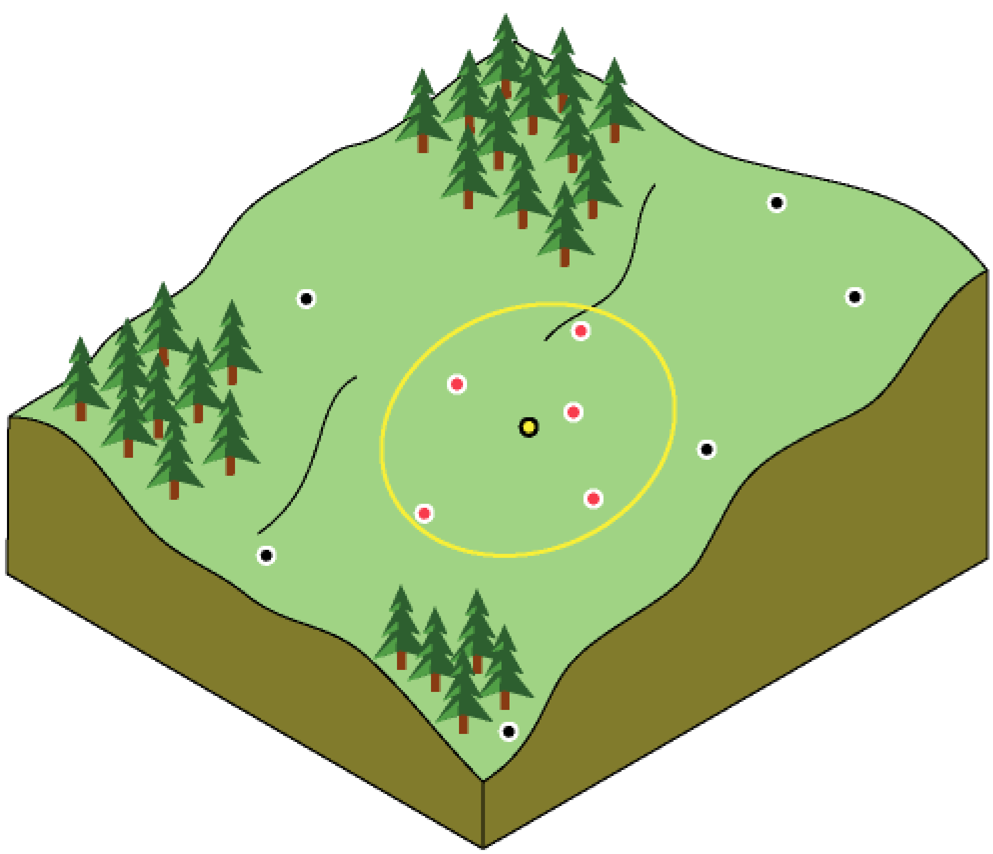


Fig 2.4 The Search Neighbourhood

CHAPTER 3
MODULE

3.1 INTRODUCTION

The VVM501 ESP32 4G LTE SIMA7672S module is a versatile and powerful embedded system module designed for wireless communication applications. It combines the capabilities of the ESP32 microcontroller with 4G LTE connectivity, providing seamless wireless communication in a compact form factor.This module boasts a range of impressive specifications and features, making it an ideal choice for projects requiring reliable and high-speed wireless communication. With its ESP32 microcontroller, the module offers dual-core processing power, Wi-Fi connectivity, Bluetooth capabilities, and robust security features.

One of the key highlights of the VVM501 module is its integrated 4G LTE modem, which enables fast and reliable cellular connectivity. This allows for remote monitoring, data transmission, and Internet access in areas without Wi-Fi coverage, making it suitable for a wide range of IoT (Internet of Things) applications.Furthermore, the module supports a variety of communication protocols, including HTTP, MQTT, TCP/IP, and more, making it compatible with various IoT platforms and cloud services. Its flexible and programmable nature allows developers to customize and tailor its functionality to suit specific project requirements.

The decision to choose the VVM501 ESP32 4G LTE SIMA7672S module for the project was based on several factors. Firstly, its powerful processing capabilities and integrated 4G LTE modem offered a comprehensive solution for wireless communication needs. Additionally, its compact size and low power consumption made it suitable for integration into the existing system architecture without requiring significant modifications. Moreover, the module's compatibility with popular development frameworks and extensive documentation provided by the manufacturer facilitated smooth integration and rapid development of firmware. Its robust security features also ensured data integrity and protection, essential for IoT applications handling sensitive information.

3.2 HARDWARE FEATURES

The VVM501 ESP32 4G LTE SIMA7672S module comprises several key hardware components and features that contribute to its versatility and performance in wireless communication applications. This section provides a detailed description of these components and features, highlighting their significance in the project's context.

ESP32 Microcontroller: At the heart of the VVM501 module lies the ESP32 microcontroller, renowned for its dual-core processing power and versatility. The ESP32 features two Tensilica LX6 CPUs clocked at up to 240 MHz, offering ample computational capabilities for handling complex tasks and data processing. This dual-core architecture allows for multitasking and efficient resource utilization, essential for responsive and scalable applications.

4G LTE Modem: The module is equipped with an integrated 4G LTE modem, enabling high-speed cellular connectivity for data transmission and Internet access. This modem supports various LTE bands and protocols, ensuring compatibility with global cellular networks. The inclusion of 4G LTE connectivity expands the module's reach and applicability, allowing for remote monitoring, asset tracking, and IoT deployments in areas with cellular coverage.

Wireless Connectivity: In addition to 4G LTE, the VVM501 module supports Wi-Fi and Bluetooth connectivity options. The built-in Wi-Fi functionality enables wireless communication over local networks, facilitating seamless integration with existing Wi-Fi infrastructure. Bluetooth support further enhances connectivity, enabling short-range communication with peripherals and mobile devices.

Memory: The module features onboard flash memory and RAM, providing ample storage and memory resources for firmware and data storage. This allows for the implementation of complex applications and data-intensive operations without compromising performance or reliability. The availability of sufficient memory resources ensures smooth operation and efficient utilization of the module's capabilities.

Peripheral Interfaces: The VVM501 module is equipped with a range of peripheral interfaces, including UART, SPI, I2C, GPIO, and more. These interfaces enable seamless integration with external sensors, actuators, displays, and other peripheral devices, expanding the module's functionality and versatility. The flexibility offered by these interfaces allows for the development of custom solutions tailored to specific project requirements.

Power Management: Efficient power management features are integrated into the module to optimize power consumption and prolong battery life in battery-operated applications. This includes low-power modes, sleep modes, and voltage regulation mechanisms to minimize power consumption during idle periods.

Form Factor and Mounting Options: The VVM501 module is designed to be compact and lightweight, making it suitable for integration into space-constrained environments and portable devices. Various mounting options, including surface mount and through-hole mounting, provide flexibility in installation and deployment, accommodating diverse application scenarios.Robustness and Reliability: The module is engineered to withstand harsh operating conditions and environments, with built-in protection against environmental factors such as moisture, temperature extremes, and electromagnetic interference. This ensures reliability and longevity in demanding industrial and outdoor applications where reliability is paramount.

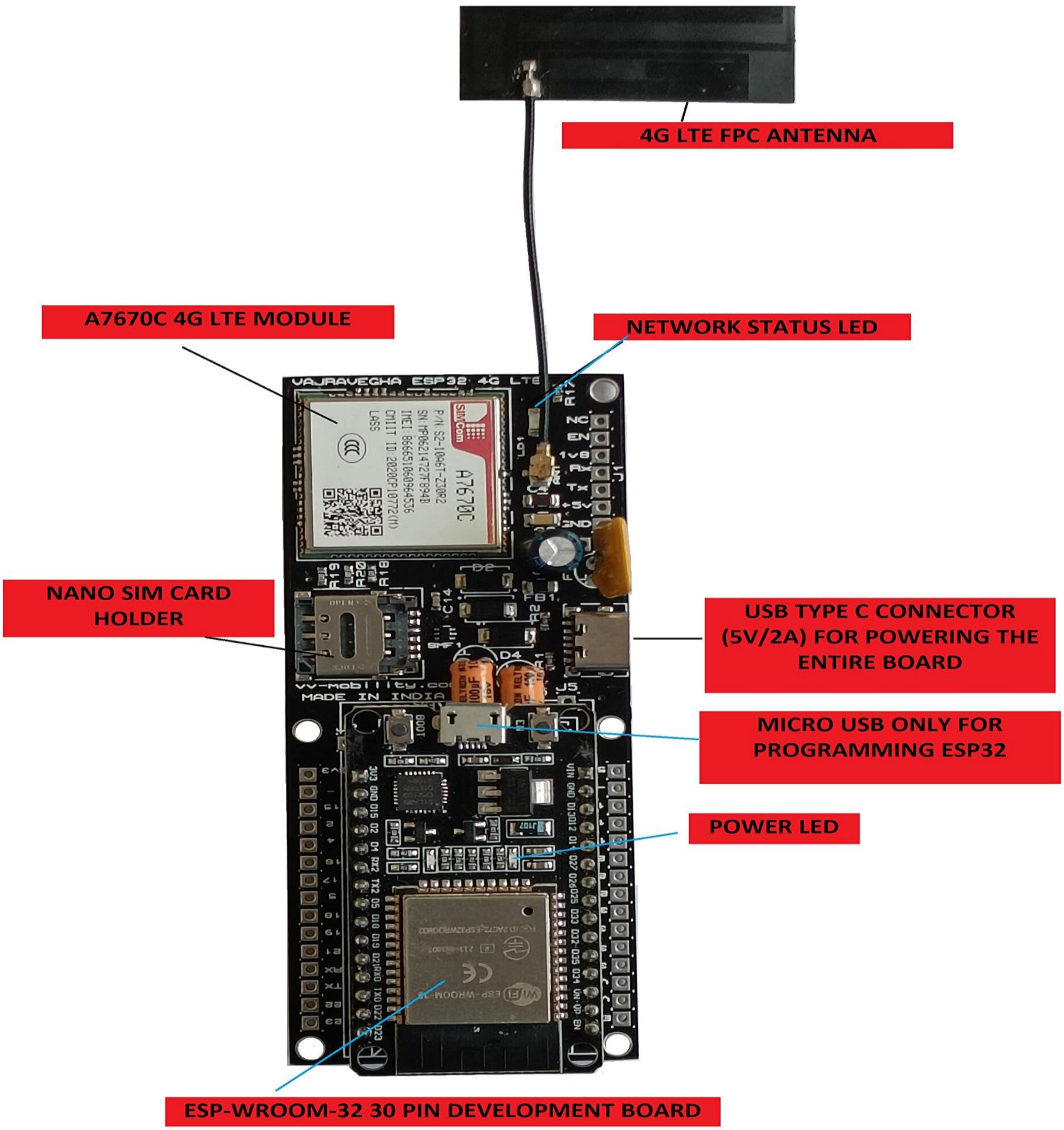


Fig 3.1 Module

3.3 SOFTWARE INTEGRATION

The VVM501 ESP32 4G LTE SIMA7672S module is a versatile and robust embedded system module designed to facilitate wireless communication in various IoT (Internet of Things) and industrial applications. This section provides an in-depth exploration of the hardware features of the VVM501 module, highlighting its key components and functionalities.At the core of the VVM501 module lies the ESP32 microcontroller, renowned for its powerful dual-core processing capabilities. Featuring two Tensilica LX6 CPUs clocked at up to 240 MHz, the ESP32 enables efficient multitasking and resource management, making it well-suited for handling complex tasks and data processing in real-time applications. The dual-core architecture ensures optimal performance and responsiveness, allowing the module to meet the demands of high-throughput data processing and communication.

One of the standout features of the VVM501 module is its integrated 4G LTE modem, which provides seamless cellular connectivity for data transmission and Internet access. Supporting various LTE bands and protocols, the 4G LTE modem enables high-speed wireless communication, making it ideal for remote monitoring, asset tracking, and telemetry applications. The inclusion of 4G LTE connectivity expands the module's reach and applicability, allowing for deployments in areas with limited or no Wi-Fi coverage.

In addition to 4G LTE connectivity, the VVM501 module offers built-in Wi-Fi and Bluetooth capabilities, providing versatile wireless communication options. The Wi-Fi functionality enables connectivity to local wireless networks, facilitating seamless integration with existing Wi-Fi infrastructure. Bluetooth support further enhances connectivity, enabling short-range communication with peripherals, mobile devices, and other Bluetooth-enabled devices.The module is equipped with onboard flash memory and RAM, providing ample storage and memory resources for firmware and data storage. This allows for the implementation of complex applications and data-intensive operations without compromising performance or reliability. The availability of sufficient memory resources ensures smooth operation and efficient utilization of the module's capabilities.

Peripheral interfaces such as UART, SPI, I2C, GPIO, and more are integrated into the VVM501 module, allowing for seamless integration with external sensors, actuators, displays, and other peripheral devices. These interfaces provide flexibility and versatility, enabling the development of custom solutions tailored to specific project requirements. By leveraging these interfaces, developers can easily interface with a wide range of sensors and actuators, expanding the module's functionality and applicability.Efficient power management features are incorporated into the VVM501 module to optimize power consumption and prolong battery life in battery-operated applications. Low-power modes, sleep modes, and voltage regulation mechanisms minimize power consumption during idle periods, maximizing energy efficiency and extending battery life. This ensures reliable operation in battery-powered devices, making the module suitable for applications requiring long-term, autonomous operation.

The compact form factor and lightweight design of the VVM501 module make it suitable for integration into space-constrained environments and portable devices. Various mounting options, including surface mount and through-hole mounting, provide flexibility in installation and deployment, accommodating diverse application scenarios. The module's robust construction and built-in protection mechanisms ensure reliability and longevity in harsh operating conditions, making it suitable for industrial and outdoor applications where durability is essential.

In the context of the project, the VVM501 module played a crucial role in enabling wireless communication capabilities for data exchange and remote monitoring. Leveraging its powerful microcontroller, integrated connectivity options, memory resources, and peripheral interfaces, the module facilitated seamless integration and deployment in the existing system architecture. By interfacing with MQTT (Message Queuing Telemetry Transport) server/client and HiveMQ Cloud, the module enabled efficient data exchange and communication between connected devices and cloud platforms. MQTT, a lightweight messaging protocol, allowed for real-time communication between devices, enabling efficient data transmission and synchronization. Integration with HiveMQ Cloud provided a scalable and reliable cloud infrastructure for data storage, analysis, and visualization, enhancing the project's capabilities for remote monitoring and management.

3.4 COMMUNICATION PROTOCOLS SUPPORTED

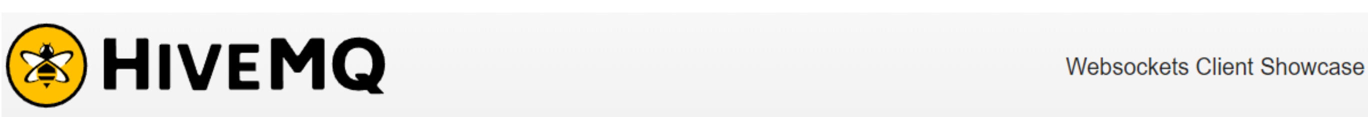
The VVM501 ESP32 4G LTE SIMA7672S module supports a range of communication protocols, including HTTP (Hypertext Transfer Protocol), MQTT (Message Queuing Telemetry Transport), and more. Each protocol serves specific purposes and offers distinct advantages in terms of data exchange and communication.

HTTP Protocol: HTTP is a widely-used protocol for communication between web servers and clients, facilitating the transfer of hypertext documents. In the context of the project, the HTTP protocol was utilized for sending and receiving data over the Internet. For example, HTTP requests were sent to retrieve data from remote servers or APIs, while HTTP responses contained data or information sent back from the server. This protocol enabled seamless integration with web-based services and APIs, allowing for real-time data retrieval and exchange.

MQTT Protocol: MQTT is a lightweight messaging protocol designed for efficient communication between devices in IoT (Internet of Things) and M2M (Machine-to-Machine) applications. It follows a publish-subscribe messaging pattern, where clients (publishers) send messages to topics, and other clients (subscribers) receive messages from subscribed topics. In the project, MQTT was utilized for real-time data exchange and communication between connected devices and cloud platforms. The module acted as an MQTT client, publishing data to specific topics and subscribing to relevant topics to receive data from other devices or servers. This protocol facilitated efficient and reliable communication, with support for quality of service (QoS) levels to ensure message delivery.

Other Protocols: In addition to HTTP and MQTT, the VVM501 module may support other communication protocols such as TCP/IP (Transmission Control Protocol/Internet Protocol), UDP (User Datagram Protocol), and more. These protocols enable communication over various network layers, providing flexibility and compatibility with different networking environments and applications. For example, TCP/IP is commonly used for reliable, connection-oriented communication, while UDP is preferred for low-latency, connectionless communication.

Integration with MQTT Server/Client and HiveMQ Cloud: The project leveraged the capabilities of the VVM501 module to interface with MQTT server/client implementations and cloud platforms such as HiveMQ Cloud. The MQTT protocol facilitated seamless communication between devices and cloud services, enabling real-time data transmission, synchronization, and remote management. The module acted as an MQTT client, connecting to an MQTT broker/server hosted either locally or in the cloud. It published data to specific topics representing different data streams or sensor readings and subscribed to topics to receive commands or configuration updates from the cloud or other devices. Integration with HiveMQ Cloud provided a scalable and reliable cloud infrastructure for data storage, analysis, and visualization. The module communicated with HiveMQ Cloud using MQTT, exchanging data and metadata for monitoring, reporting, and control purposes. This integration enhanced the project's capabilities for remote monitoring, management, and decision-making, leveraging the power of cloud computing and IoT technologies.



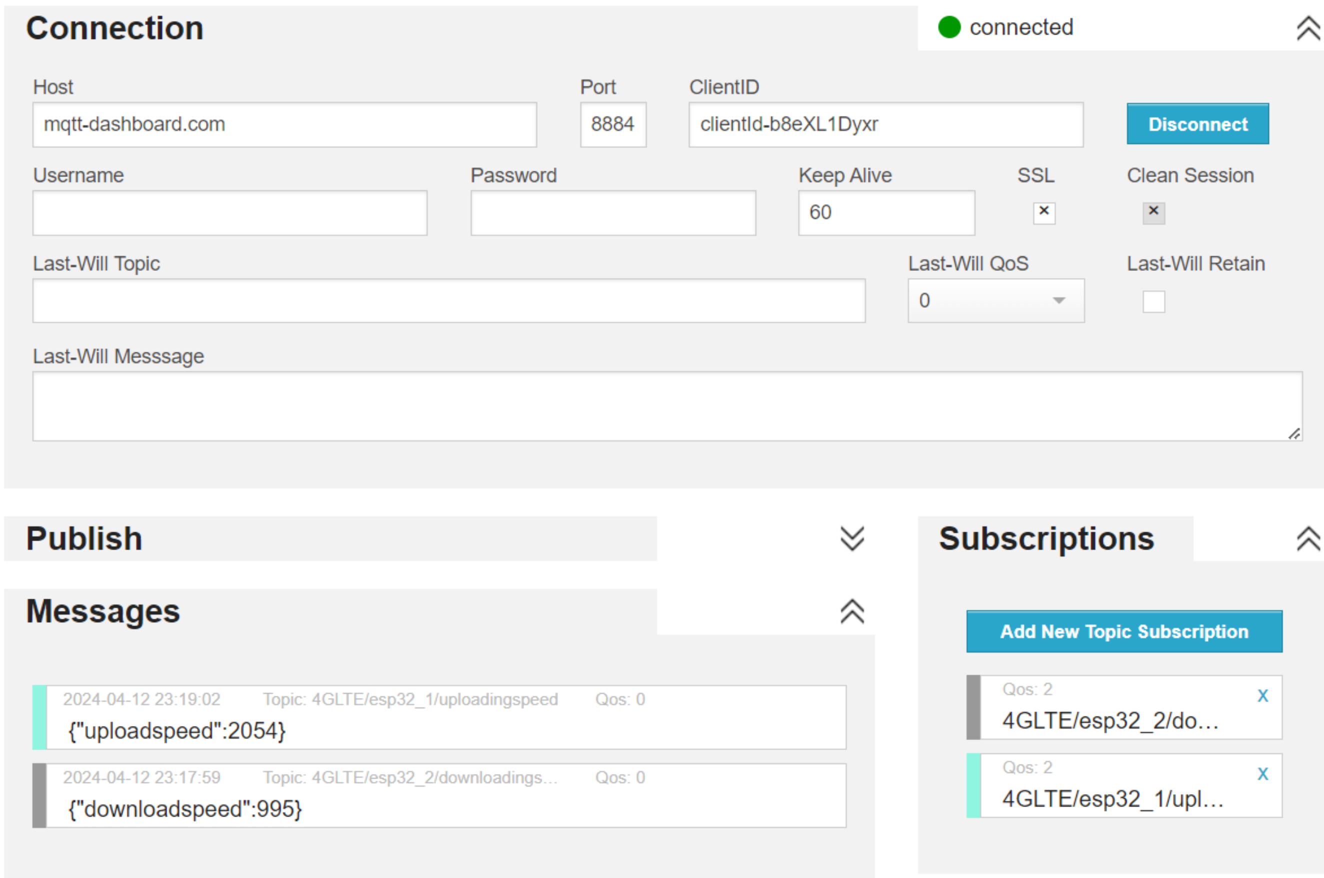


Fig 3.2 HIVEMQ Interface

Benefits of Protocol Support: By supporting multiple communication protocols such as HTTP and MQTT, the VVM501 module offered flexibility and interoperability, enabling seamless integration with existing systems, networks, and cloud services. These protocols facilitated efficient data exchange, communication, and collaboration between connected devices, servers, and cloud platforms, empowering the project with real-time monitoring, control, and automation capabilities. The modular and extensible nature of the module's communication capabilities allowed for scalability and adaptability, making it suitable for a wide range of IoT, industrial, and consumer electronics applications. Overall, the support for diverse communication protocols played a crucial role in enhancing the project's functionality, connectivity, and effectiveness in achieving its objectives.

3.5 PERFORMANCE EVALUATION OF THE VVM501 ESP32 4G LTE SIMA7672S MODULE

The performance evaluation of the VVM501 ESP32 4G LTE SIMA7672S module involved assessing various aspects such as speed, reliability, power consumption, and compatibility with different communication protocols. One key metric evaluated was the data transmission speed over the 4G LTE network, which determines the module's efficiency in sending and receiving data packets. Additionally, the reliability of the module's connection to the network was assessed by measuring the frequency of dropped connections and packet loss.

Another important aspect evaluated was the power consumption of the module during different operating conditions, including idle, standby, and active states. This involved measuring the module's current consumption and analyzing its energy efficiency, especially in battery-operated applications where power consumption is critical. Additionally, the compatibility of the module with various communication protocols, such as MQTT and HTTP, was evaluated to ensure seamless integration with existing systems and networks.

Furthermore, the module's responsiveness and latency in processing data packets and executing commands were assessed to determine its real-time performance. This involved measuring the time taken for the module to respond to incoming requests or commands and evaluating its ability to handle concurrent connections and data streams efficient

CHAPTER 4

FABRICATION

The purpose of this project was to design and fabricate a 3D-printed enclosure to house a module securely. The enclosure needed to be durable, functional, and aesthetically pleasing. The enclosure was designed using CAD software, considering the dimensions of the module and the necessary space for connectors and ventilation. The design included a hinged lid for easy access to the module and mounting holes for installation. The enclosure was fabricated using PLA (Polylactic Acid) filament, a commonly used material for 3D printing due to its strength and durability. PLA is also environmentally friendly, making it a suitable choice for this project. The completed enclosure was tested to ensure it fit the module correctly and provided adequate protection. The hinged lid was tested for ease of opening and closing, and the mounting holes were tested for secure installation.

The 3D-printed enclosure successfully met the design requirements and provided a durable and functional housing for the module. The use of PLA filament ensured that the enclosure was strong and environmentally friendly. Overall, the project was a success, demonstrating the capabilities of 3D printing for fabricating custom enclosures.

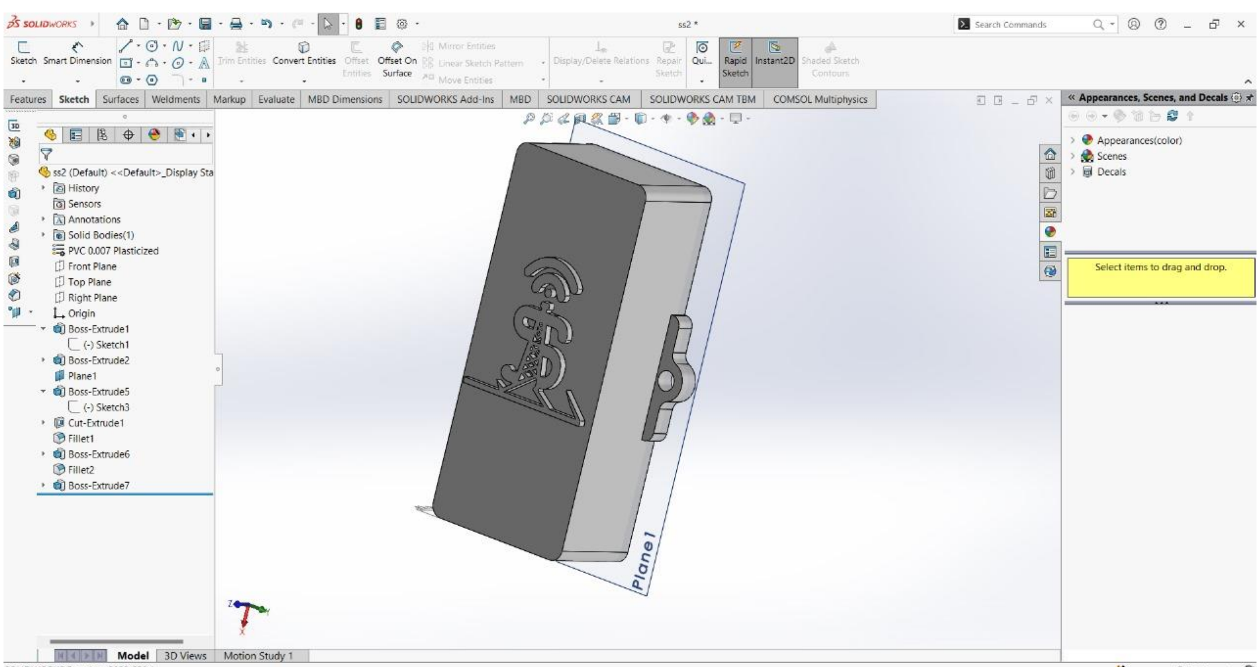


Fig 4.1 Box design

Fig 4.2 Printed box

CHAPTER 5

DATA RETRIEVAL

5.1 INTRODUCTION

The ESP32 GSM module is employed in this project to capture signal speed data. The collected data is then transmitted to a server using the MQTT (Message Queuing Telemetry Transport) protocol via the HiveMQ websocket client. To retrieve and display the transmitted data, a paho JavaScript library is used. This module aims to provide a seamless and efficient method for collecting, transmitting, and displaying signal speed data.

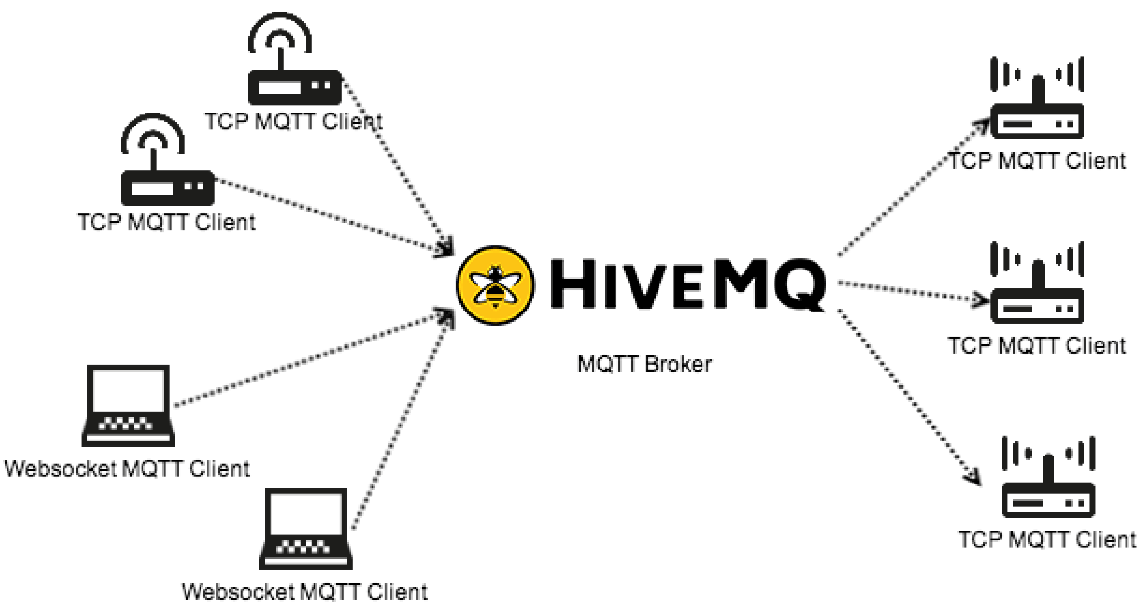
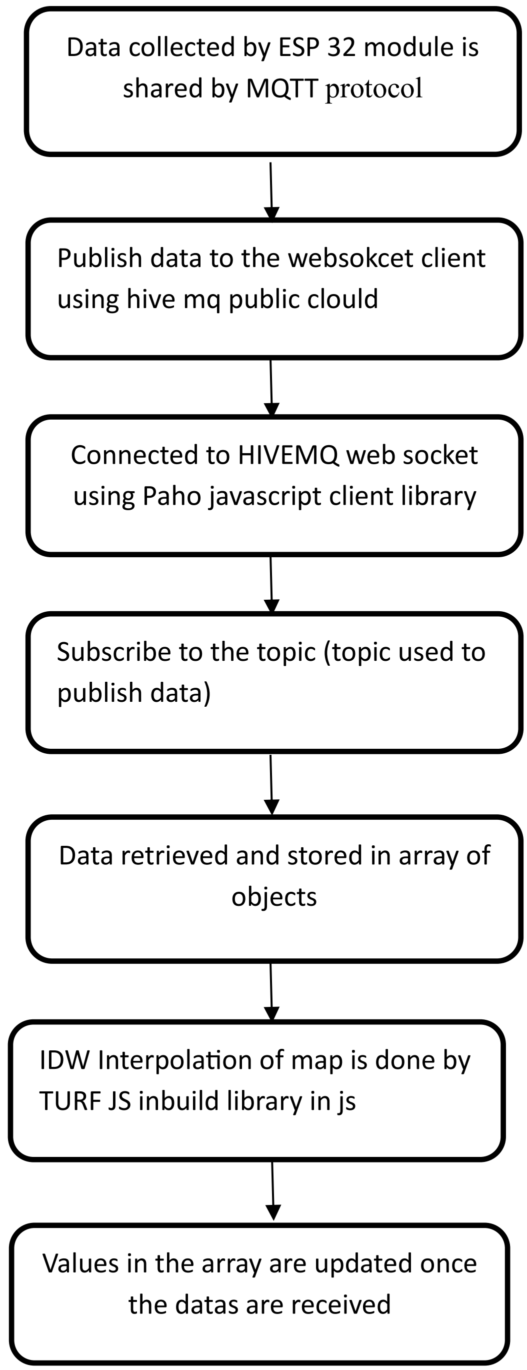


Fig 5.1 Inroduction

5.2 FLOW OF DATA SEND AND RETRIEVE METHOD



5.3 MQTT AND DEPENDENCIES

MQTT

MQTT is the most commonly used messaging protocol for the Internet of Things (IoT). MQTT stands for MQ Telemetry Transport. The protocol is a set of rules that defines how IoT devices can publish and subscribe to data over the Internet. MQTT is used for messaging and data exchange between IoT and industrial IoT (IIoT) devices, such as embedded devices, sensors, industrial PLCs, etc.

HIVEMQ WEBSOCKET CLIENT

HiveMQ also offers its WebSocket client, which is based on the Eclipse Paho MQTT JavaScript client. This client can be used to connect to WebSocket-enabled MQTT brokers, such as HiveMQ Cloud, on port 8884. The client supports SSL/TLS for secure communication, ensuring that all data transmitted between the MQTT client and the broker is encrypted and secure

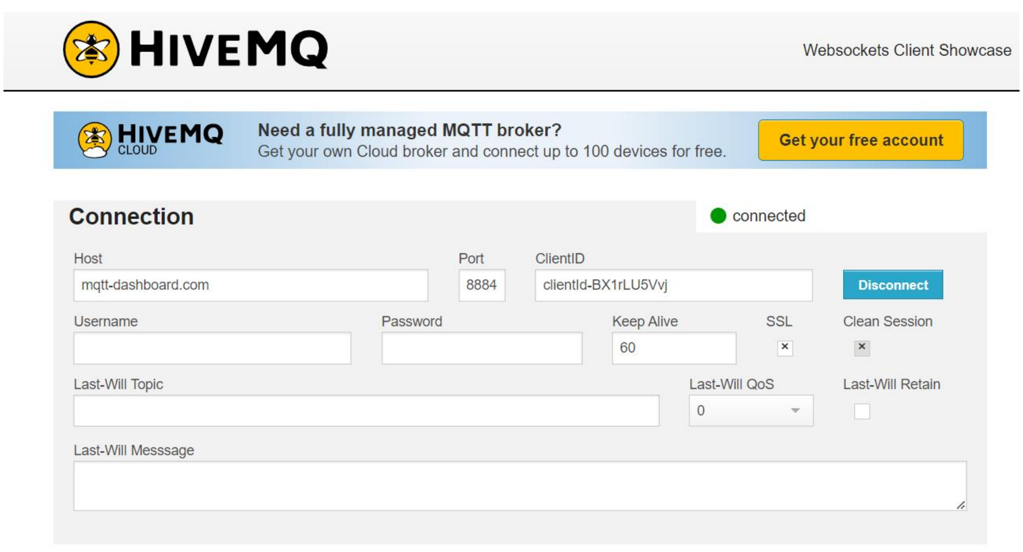


Fig 5.2 HIVE MQ Web Socket Client

MQTT BROKER

MQTT brokers receive published messages and dispatch the message to the subscribing MQTT clients. An MQTT message contains a message topic that MQTT clients subscribe to and MQTT brokers use these subscription lists for determining the MQTT clients to receive the message. The MQTT broker filters all incoming messages and distributes them correctly to the Subscribers.

PAHO JAVASCRIPT CLIENT LIBRARY:

The Eclipse Paho MQTT JavaScript client is a widely used and stable library for MQTT communication in web applications. It provides a convenient way to connect, publish, and subscribe to MQTT messages directly from web browsers using WebSockets.

SSL COMMUNICATION

SSL, or Secure Sockets Layer, is an encryption-based Internet security protocol. SSL encrypts data that is transmitted across the web. This means that anyone who tries to intercept this data will only see a garbled mix of characters that is nearly impossible to decrypt. SSL is the predecessor to the modern TLS encryption used today.

5.4 CONNECTION TO WEB SOCKET

STEPS:

1. Create an MQTT client instance using the Eclipse Paho MQTT JavaScript client.

```
const brokerEndpoint = 'mqtt-dashboard.com';
const port = 8884;
const path = '/mqtt';
const clientID = 'signal';

const client = new Paho.MQTT.Client(brokerEndpoint, port, path, clientID);

client.onConnectionLost = (responseObject) => {
    if (responseObject.errorCode !== 0) {
        console.log('Connection lost:', responseObject.errorMessage);
    }
};
```

Fig 5.3 MQTT CLIENT INSTANCE

2. Defines a set of MQTT topics to subscribe to.

```
const topics = {
    'topic1': '4GLTE/esp32_1/uploadingspeed1',
    'topic2': '4GLTE/esp32_2/downloadingspeed1',
    'topic3':'4GLTE/esp32_1/uploadingspeed2',
    'topic4':'4GLTE/esp32_2/downloadingspeed2',
    'topic5':'4GLTE/esp32_1/uploadingspeed3',
    'topic6':'4GLTE/esp32_2/downloadingspeed3',
    'topic7':'4GLTE/esp32_1/uploadingspeed4',
    'topic8':'4GLTE/esp32_2/downloadingspeed4'
};
```

Fig 5.4SET OF MQTT TOPICS TO SUBSCRIBE

3. Sets MQTT connection options with SSL/TLS enabled and authentication credentials.

```javascript
const connectOptions = {
    useSSL: true,
    userName: "",
    password: "",
    onSuccess: () => {
        console.log('Connected to MQTT broker');
        for (const key in topics) {
            if (topics.hasOwnProperty(key)) {
                const topic = topics[key];
                client.subscribe(topic);
                console.log(`Subscribed to ${topic}`);
            }
        }
    },
    onFailure: (message) => {
        console.log('Connection failed:', message.errorMessage);
    }
};
```

Fig 5.5 MQTT CONNECTION OPTIONS WITH SSL/TLS ENABLED

4. Sets event handlers for connection loss and connection success/failure.

```javascript
client.onMessageArrived = (message) => {
    console.log('Message received:', message.payloadString);
}
```

Fig 5.6 EVENT HANDLERS FOR CONNECTION LOSS AND CONNECTION SUCCESS/FAILURE

5. Establishes a connection to HiveMQ Cloud and subscribes to the defined MQTT topics.

```
Connected to MQTT broker
Subscribed to 4GLTE/esp32_1/uploadingspeed1
Subscribed to 4GLTE/esp32_2/downloadingspeed1
Subscribed to 4GLTE/esp32_1/uploadingspeed2
Subscribed to 4GLTE/esp32_2/downloadingspeed2
Subscribed to 4GLTE/esp32_1/uploadingspeed3
Subscribed to 4GLTE/esp32_2/downloadingspeed3
Subscribed to 4GLTE/esp32_1/uploadingspeed4
Subscribed to 4GLTE/esp32_2/downloadingspeed4
Message received: {"uploadspeed2":768}
Message received: {"downloadspeed2":4332}
Message received: {"uploadspeed3":470}
Message received: {"downloadspeed3":3942}
Message received: {"uploadspeed4":643}
Message received: {"downloadspeed4":3799}
Message received: {"uploadspeed1":802}
Message received: {"downloadspeed1":4144}
```

Fig 5.7 Output of the message received

6. Real-Time Updation

Data is collected from specific locations, with each location corresponding to a different topic on the ESP32 module. Based on the topic, the values in the array are updated for that particular location. The array of data is updated based on the received MQTT messages and performs interpolation to update the maps with the new data.

CHAPTER 6

ESTIMATED SIGNAL STRENGTH

6.1 INTRODUCTION

Since it is a **estimation** of the **signal strength** at all the points inside the college with respect to the data obtained by modem and some extra stationary control points.

The difference between the two is that for the ground control the data will be a time averaged signal data for a particular network provider. Whereas the modem will be detecting the live signal strength and send it to the server, so it will be changing with the time. The reason for choosing the stationary control points is just to cover the various other locations in the campus whose data may impact the primary signal strength map for the entire campus;

However, some of the interpolation techniques used for achieving this includes;

- **Kriging interpolation;**
- **IDW interpolation;**

IDW interpolation is however a sub set of kriging which uses much more intense algorithm;

Here the various methods to achieve the result has been explained below;

6.2 ALGORITHM 1

However, the **algorithm 1** used for producing the output has been described below:

The main algorithm used in the provided python code performs the following tasks:

Import libraries: import the necessary libraries for geospatial analysis and visualization which includes geopandas, pandas, matplotlib, numpy and etc.

Load data: load a csv file containing signal strength data and a geojson file containing boundary information which is the shapefile of our college.

Data preparation:

- Convert latitude and longitude columns from the csv file into point geometries.
- Create a geodataframe from the csv data with point geometries.

Spatial join: perform a spatial join between the geodataframe created from the csv data and the geodataframe loaded from the geojson file. This associates each point with the boundary it intersects.

Data export: export the merged data to a csv file.

Interpolation:

- Define a grid for interpolation based on the bounding box of the boundary.
- Use ordinary kriging interpolation to estimate values at grid points based on observed values from the merged data.

Geodataframe creation:

- Create polygon geometries representing the grid cells.
- Create a geodataframe from these polygon geometries.

Data manipulation:

- Add the interpolated values as a new column in the geodataframe.
- **Export interpolated data**: save the interpolated geodataframe as a geojson file.

Visualization:

- Visualize the interpolated values using **plt.imshow()**.
- Overlay the boundary shapefile on the plot.
- Add a color bar to the plot for reference.
- Display the plot.

Overall, the algorithm integrates geospatial data processing (loading, manipulation, and analysis) with interpolation techniques and visualization to provide insights into the distribution of signal strength within a geographical area.

The detailed code with explanation has been attached in the github link; https://github.com/swattik0487/ctf/blob/main/spatial%20(5).p

OUTPUT

However the output and its visuability will be highly dependent on the colour attributes and significantly on the number of pixels in the image. If the image contains higher number of pixels it has more clarity about the data of each point in the campus and subsequently it will consume more data while image compilation and display on the web.Here the output generated contains grid of 30×30 which can be modifed to 40×40 or even 60×60 for a better clarity data.

There are various sequential colour maps which we can modify by just changing "cmap" attributes.

1. For cmap = plasma and the grid is 30×30.

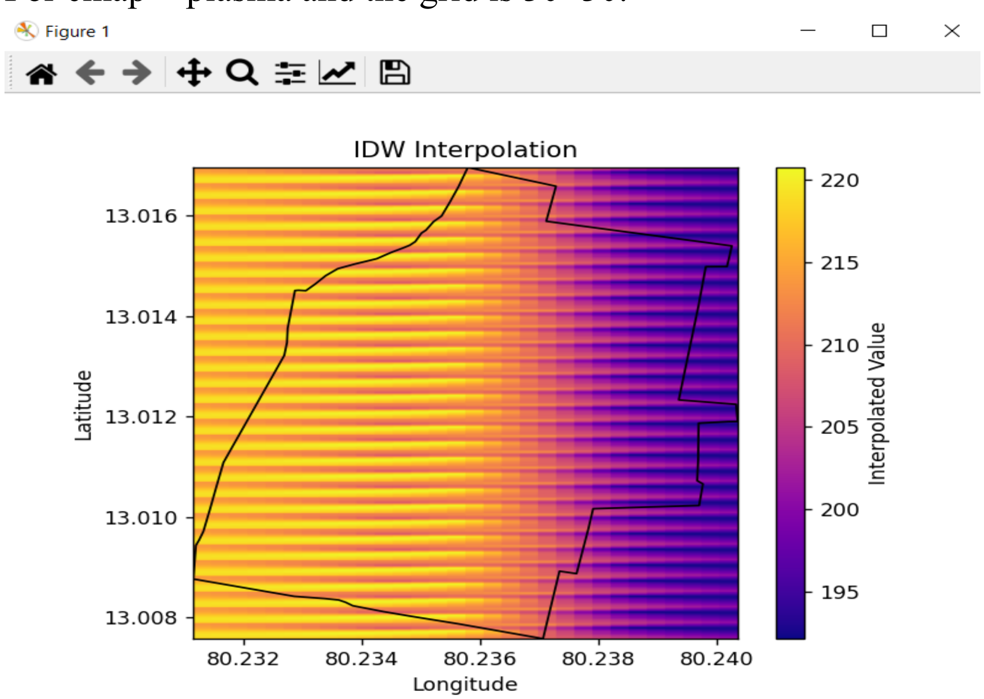
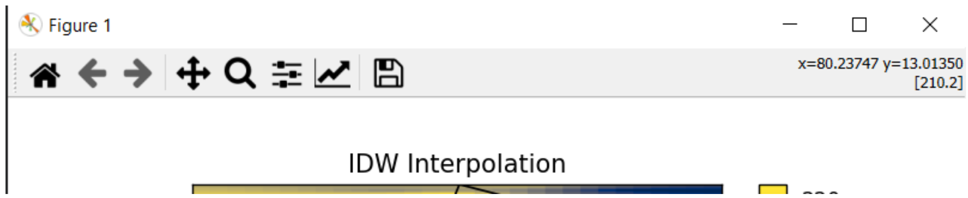


Fig 6.1 NO. 1 OUTPUT OF ALGORITHM 1

2. For cmap = virdris and the grid is 30×30

Here we are just changing the colour palette denoting various signal strength values. Also if we move the cursor over the map we can generate its x, y values as well as its estimated signal values.



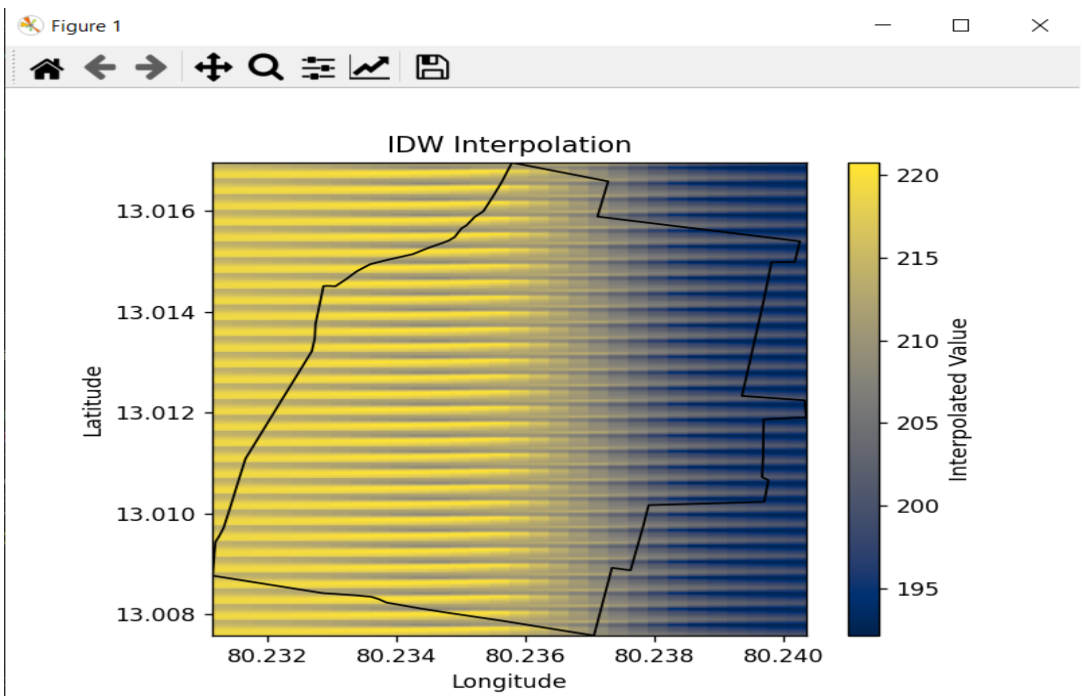


Fig 6.2 NO. 2 OUTPUT OF ALGORITHM 1

Now coming to the second algorithm where the interpolation where the same concept is being used. In additional we are trying to do web display of the interpolated layer by using flask library in python.

6.3 ALGORITHM 2

This flask application serves an interpolated layer as a geojson endpoint based on provided shapefile and csv data. Here's the algorithm for the provided code:

Initialization:

- Import necessary libraries (**flask**, **render_template**, **jsonify**, **geopandas**, **pandas**, **numpy**, **scipy.interpolate.rbf**).
- Initialize flask application.

Routes:

- Define a route **'/'** which renders **index1.html**.
- Define a route **'/interpolated_layer.geojson'** which returns the interpolated layer as geojson data.

Interpolation function (**interpolate_shapefile_and_csv**):

- Load the shapefile and csv data.
- Convert the csv data to a geodataframe.
- Perform a spatial join between the shapefile and the csv data.
- Extract coordinates and values from the merged data.
- Perform interpolation using inverse distance weighting (IDW) with the **IDW_interpolation** function.
- Create a new geodataframe containing the interpolated values.

IDW interpolation function (**idw_interpolation**):

- Utilizes the **scipy.interpolate.rbf** function to perform inverse distance weighting interpolation.

Main block:

- Run the flask application on **127.0.0.1:5000**.

This code assumes the following:

- The shapefile is in GeoJson format.
- The csv file contains columns named **'longitude'**, **'latitude'**, and **'value'**.
- Both shapefile and csv data have valid geometries.
- The interpolation method used is inverse distance weighting (idw).

The detailed code has been attached in the attached in the github link:

Python file:https://github.com/swattik0487/ctf/blob/main/file1%20(4).py

Html file:https://github.com/swattik0487/ctf/blob/main/index1.html

Csv file :
https://github.com/swattik0487/ctf/blob/main/signal%20strength%20up.csv

geojson file: https://github.com/swattik0487/ctf/blob/main/ceg_boundary.geojson

OUTPUT

Fig 6.3 NO. 1 OUTPUT OF ALGORITHM 2

The interpolated layer is displayed here as a cluster of several interpolated points overlayed over the base map extracted by using leaflet whose view properties has already been set in the index1.html file. If we want the view can be modified by changing the attributes.

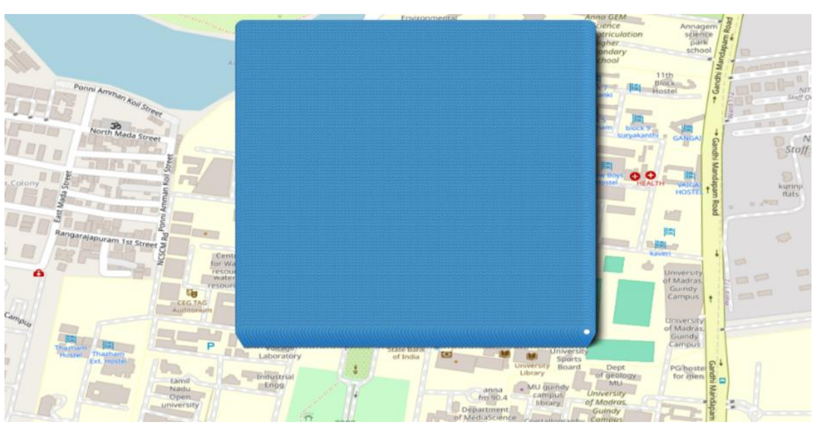


Fig 6.4 NO. 2 OUTPUT OF ALGORITHM 2

Now the third algorithm for doing interpolation is by using qgis[an open source geographic information system software] plugin for doing the interpolation and running it in the web.

6.4 ALGORITHM 3

Here the various shape files and their properties are set in the qgis and also attached with the csv file which contains the attribute data.

Then to make the map much more visually appealing an animated layer of the map is attached underneath by using the qgis plugins like **quickmapservices** and then **osm[open street map],** also in the same popup we can see the **qgis2web** used for the web display:

OUTPUT

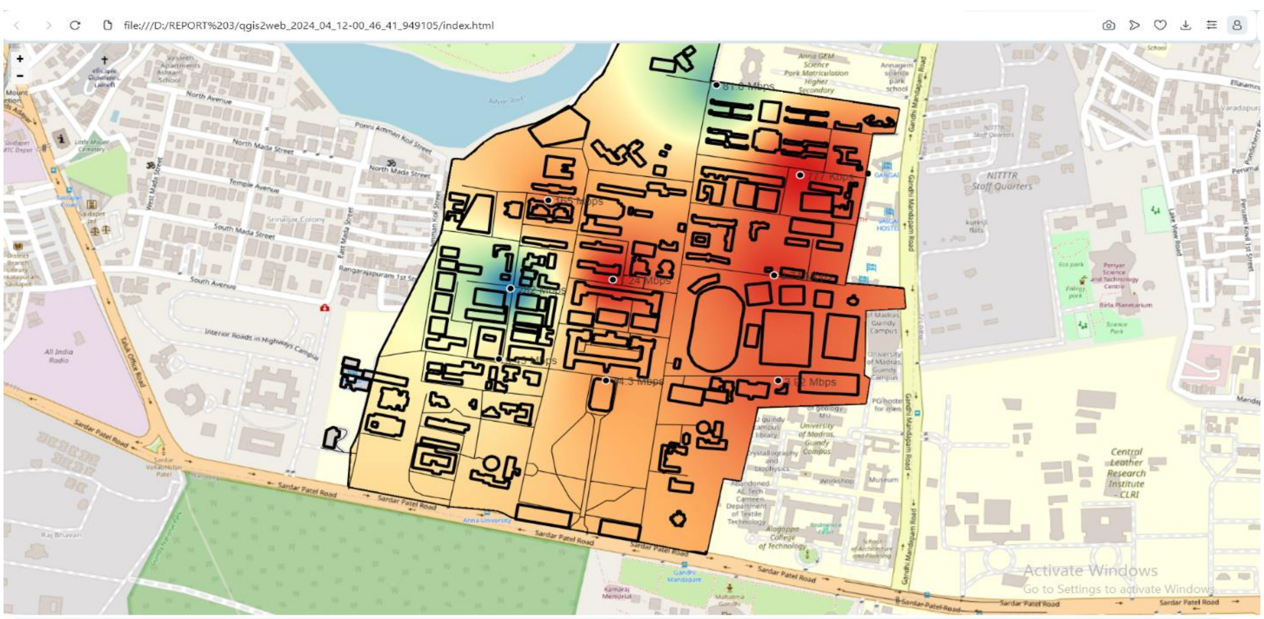


Fig 6.5 OUTPUT OF ALGORITHM 3

Also we want to perform further modication to the result it can be easily done by extracting the source code from qgis and manipulating the source code;

```
qgis2web_2024_04_12-00_46_41_949105 > index.html > html > head > style > html > header
<!doctype html>
<html lang="en">
    <header> SIGNAL SAFARI; AN INITIATIVE BY CEG TECH FORUM</header>
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="initial-scale=1,user-scalable=no,maximum-scale=1,width=device-width">
        <meta name="mobile-web-app-capable" content="yes">
        <meta name="apple-mobile-web-app-capable" content="yes">
        <link rel="stylesheet" href="./resources/ol.css">
        <link rel="stylesheet" href="resources/fontawesome-all.min.css">
        <link rel="stylesheet" href="./resources/ol-layerswitcher.css">
        <link rel="stylesheet" href="./resources/qgis2web.css">
<style>
.search-layer {
  top: 65px;
  left: .5em;
}
.ol-touch .search-layer {
```

Fig 6.6 SOURCE CODE FROM QGIS FOR ALGORITHM 3

Modified output:

Fig 6.7 MODIFIED OUTPUT OF ALGORITHM 3

however, the detailed source code is attached in the github link;

https://github.com/swattik0487/ctf/blob/main/index.html

Also the related files and the resources such as the shape files and etc. Are attached in this google drive folder;

https://drive.google.com/drive/folders/1uhj7koayogiw8txdev8k1lkoojz-ds9g?usp=drive_link

6.5 INTERPOLATION USING JAVASCRIPT

After the data is published on the subscribed topics, the messages are retrieved in JSON (JavaScript Object Notation) format which is parsed to convert into JavaScript objects. Signal speed from the messages are stored in an array of objects, along with the location name, latitude, and longitude. This array of objects is then used for IDW interpolation using the TURF JS library.

Turf.js is an open-source JavaScript library used for spatial analysis. It includes traditional spatial operations, helper functions for creating GeoJSON data, and data classification and statistics tools. Turf can be added to your website as a client-side plugin, or you can run Turf server-side with Node.js. Interpolation is the process of estimating the value of a variable at locations where it has not been directly measured, based on the known values at surrounding sample points. The estimated values are used to create a continuous surface that can be used for mapping and analysis purposes. There are various interpolation tools available in the Turf.js library out of which we have chosen to use Inverse Distance Weight (IDW) because of its wide acceptance and effectiveness. Inverse distance weighted (IDW) interpolation determines cell values using a linearly weighted combination of a set of sample points. The weight is a function of inverse distance. The surface being interpolated should be that of a location dependent variable. One of the major advantages of using JavaScript libraries is that dynamic actions can be easily performed within the website. In this case the data retrieval and interpolation happens dynamically. In this project the download and upload speed data that has been retrieved from the server is being interpolated using turf.js. Since there is only 4 dynamic point data is available we have collected 8 additional point's download and upload speed manually and have also used them for the interpolation for more accurate results. One of the main input parameters is the cell size of the interpolation. We found the any cell size below 0.3 was computationally complex and took a long time to process. So we fixed with the cell size of 0.3. Later we classified the interpolation into 4 classes based on their value to differentiate regions. We used different colour codes to differentiate those regions for better visual understanding. Finally a legend was prepared and added adjacent to the maps for better interpretation.

FINAL OUTPUT OF THE INTERPOLATED MAP:

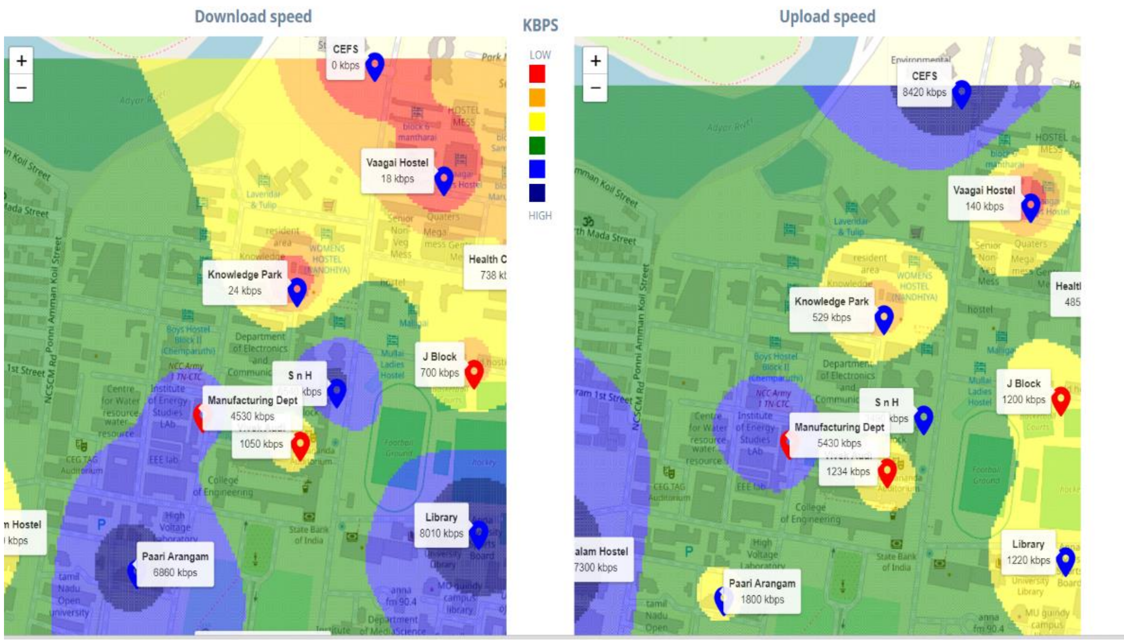


Fig 6.8 FINAL OUTPUT OF THE INTERPOLATED MAP

6.6 ANALYSIS

After going through these three processes we have analysed thwe correctness and the accuracy of each of these layers by comparing it with the ground data.

In that , we found that the 1st method is not able to generate high accuracy while 2nd method though providig us with accurate interpolated data but is taking too much time to run and provide visual display with a suitable colour palette and the 3rd method though fulfilling all these issues is unable to hangle dynamic data under a function.

Hence we move with **turf.js** and **leaflet** libraries in the javascript and aim to do the interpolation there whose basis can be explained here as;

Here first the points are added with their markers on the base structured by the leaflet library and then its converted to turf point layer where the interpolation attribute like the grid properties and colour pallete are added

Example of final output:

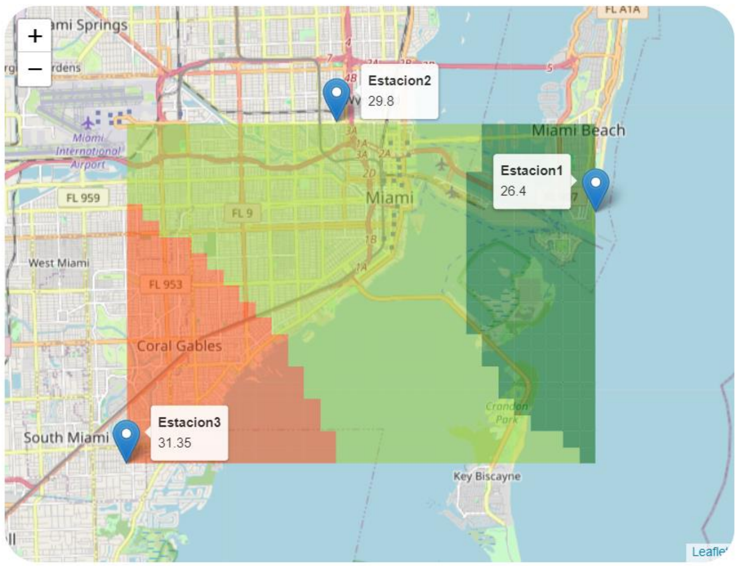


Fig 6.9 OUTPUT EXAMPLE

CHAPTER 7

FRONT END DEVELOPMENT

7.1 INTRODUCTION

Front-end development, also known as client-side development, is the process of designing and building the user interface (UI) and user experience (UX) of a website or web application. It involves creating the visible parts of a website that users interact with directly in their web browsers. It is the process of creating and implementing the visual and interactive elements of websites and web applications. Front-end development involves using web technologies like HTML, CSS, and JavaScript to build the user interface and user experience of a website. It encompasses a range of web technologies like HTML, CSS, and JavaScript, tools, and practices to ensure that web applications are visually appealing, functional, and easy to use. It is considered one of the main visual outputs of the project in which the dynamic map was to be embedded, which changes automatically with the real-time data from the sensors.

HTML (Hypertext Markup Language):

HTML is the standard markup language used to structure the content of web pages. It provides the basic building blocks for creating web pages by defining elements such as headings, paragraphs, lists, links, images, and forms. Front-end developers use HTML to organize and represent the information on a web page in a structured and semantic manner.

CSS (Cascading Style Sheets):

CSS is a styling language used to control the presentation and layout of HTML elements on a web page. It allows front-end developers to define styles such as colors, fonts, spacing, borders, and animations to enhance the visual appearance of a website. CSS enables developers to create consistent and visually appealing user interfaces across different devices and screen sizes.

JavaScript:

JavaScript is a programming language used to add interactivity and dynamic behavior to web pages. It allows front-end developers to create interactive features such as form validation, animations, event handling, and DOM manipulation. JavaScript enhances the user experience by making web applications more responsive and engaging.

This phase of the project was subdivided into various stages which include:

1) Requirement Gathering
2) Front-end Architecture Planning
3) Development and testing
4) Review and Feedback
5) Deployment

7.2 REQUIREMENT GATHERING

It includes understanding the project objectives, target audience, and their requirements. The main objective of the project is to assess signal strength from key cellular operators across the college campus. Its focus is on creating a dynamic, automated mapping system that visually illustrates real-time network coverage.
The main target audience is the students of CEG who have the prime requirement of locating areas inside the campus with good signal strength.

The front end of the website was to be designed with the main focus on the map, giving additional information about its working along with it.

7.3 FRONT-END ARCHITECTURE PLANNING

- The next phase includes choosing appropriate frameworks, libraries, and tools and determining the organization of code, file structure, and component hierarchy for the front end.
- The initial test case of the website was designed in an online web designing and hosting platform, Webflow which offers high-end web development tools with strong, responsive design.
- Technology stack:
    - HTML
    - CSS
    - Javascript
    - Webflow

- Atomic design principles and box model have been decided to organize components into atoms, molecules, organisms, templates, and pages.

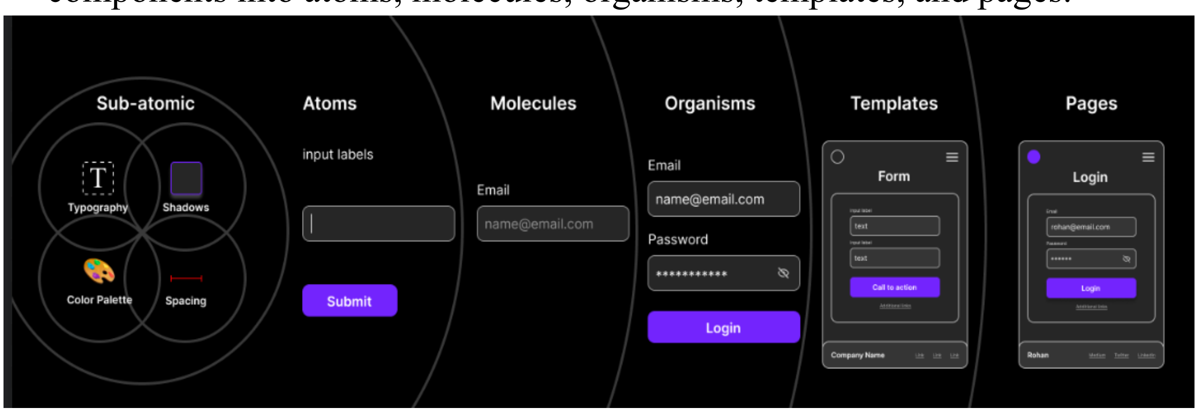
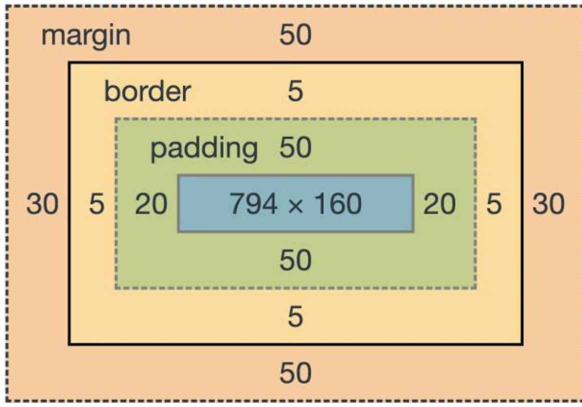
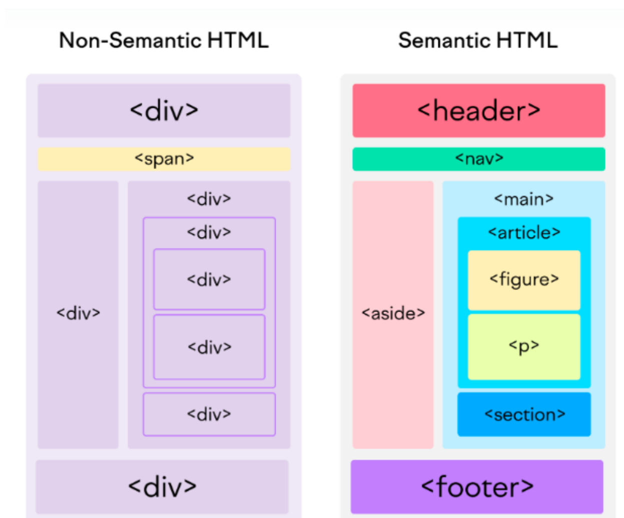


Fig 7.1 FRONT-END ARCHITECTURE PLANNING



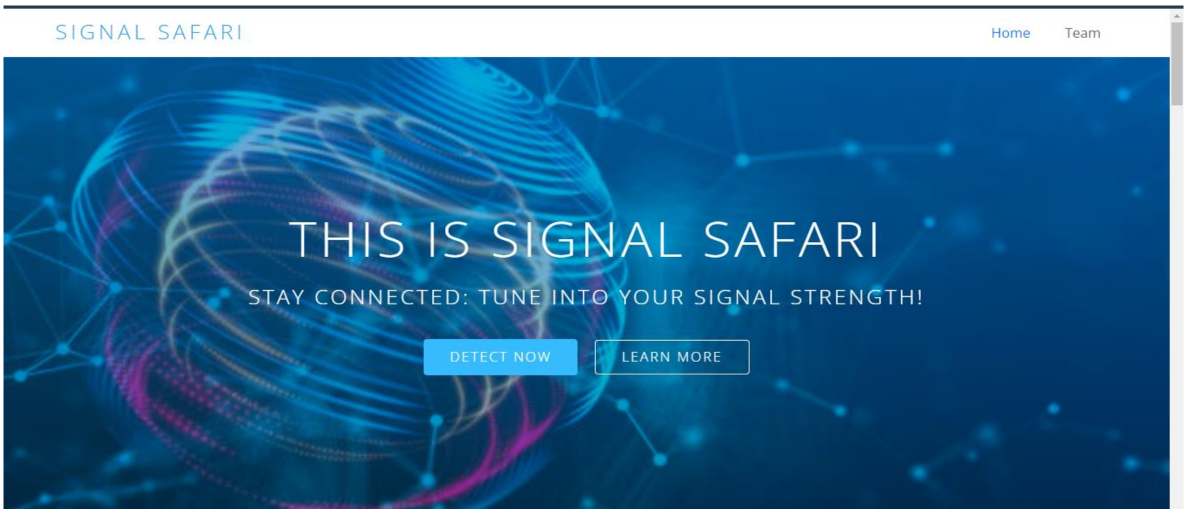
- Semantic markup <header>, <article>, and <footer> are to be implemented to improve readability and indicate the role of the content they contain even for screen readers



7.4 DEVELOPMENT AND TESTING

- The front end has been developed on the Webflow platform taking a standard template and customizing it based on the requirement using HTML, CSS, and Javascript.

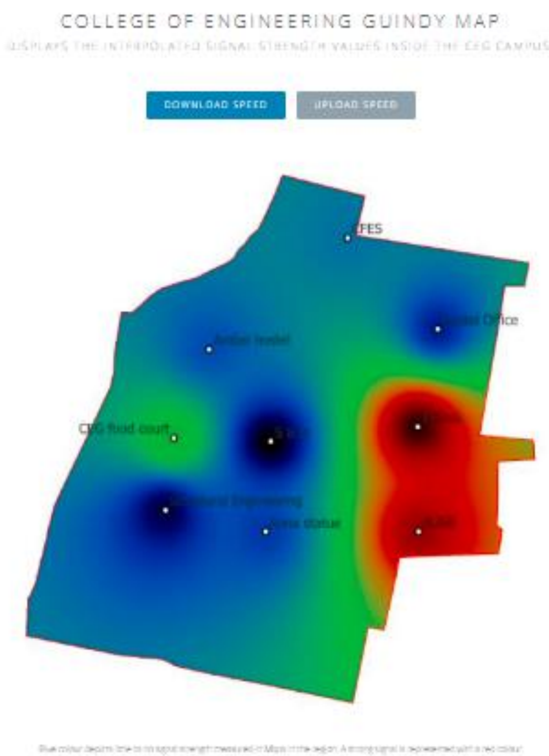
- It has been divided into several section elements each carrying a specific description of the features that have been included in the map. These sections include:
    - A “hero section” that gives a general idea of the project (i.e.) signal safari.
        - A navbar that has internal links (hyperlinks) to all the sections of the website
        - 2 buttons and a span element of “SIGNAL SAFARI” in the navbar

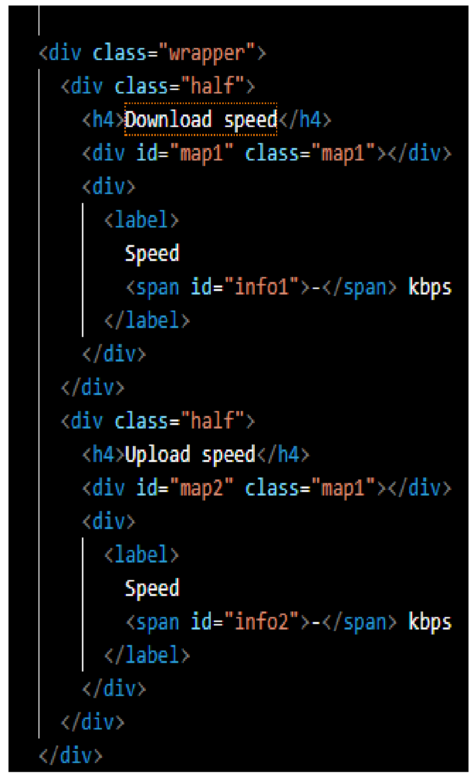



```html
<div data-collapse="medium" data-animation="default" data-duration="400" data-easing="ease" data-easing2="ease"
    role="banner" class="navigation-bar w-nav">
    <div class="w-container"><a href="#hero-section" class="brand-link w-nav-brand">
            <h1 class="brand-text">Signal safari</h1>
        </a>
        <nav role="navigation" class="navigation-menu w-nav-menu"><a href="#hero-section"
                class="navigation-link w-nav-link">Home</a><a href="#contact"
                class="navigation-link w-nav-link">Team</a></nav>
        <div class="hamburger-button w-nav-button">
            <div class="w-icon-nav-menu"></div>
        </div>
    </div>
</div>
<div id="hero-section" class="hero-section centered">
    <div data-w-id="e464d218-f801-55d1-1f50-7da00b5bfb8f" style="opacity:0" class="w-container">
        <h1 data-ix="fade-in-bottom-page-loads" class="hero-heading">This is signal safari</h1>
        <div data-ix="fade-in-bottom-page-loads" class="hero-subheading">Stay Connected: Tune into Your Signal
            Strength! </div>
        <div data-ix="fade-in-bottom-page-loads"><a href="#Detect-Now" class="button-team">Detect now</a><a
                href="#learn-more" class="hollow-button all-caps">Learn more</a></div>
    </div>
</div>
```

Fig 7.2 WEBPAGE AND HTML CODE - 1

- Another important section includes the area where the dynamic map is inserted and all the changes have been done. For testing the static map developed in Phase 1 was used as a placeholder to check and estimate the rough side of the dynamic map that was to be inserted.
- It consists of a heading element <h2></h2> followed by two buttons <button></button> elements which are styled using CSS that act as a switch window case between the upload and download speed tabs as the snippet of the code attached following this:

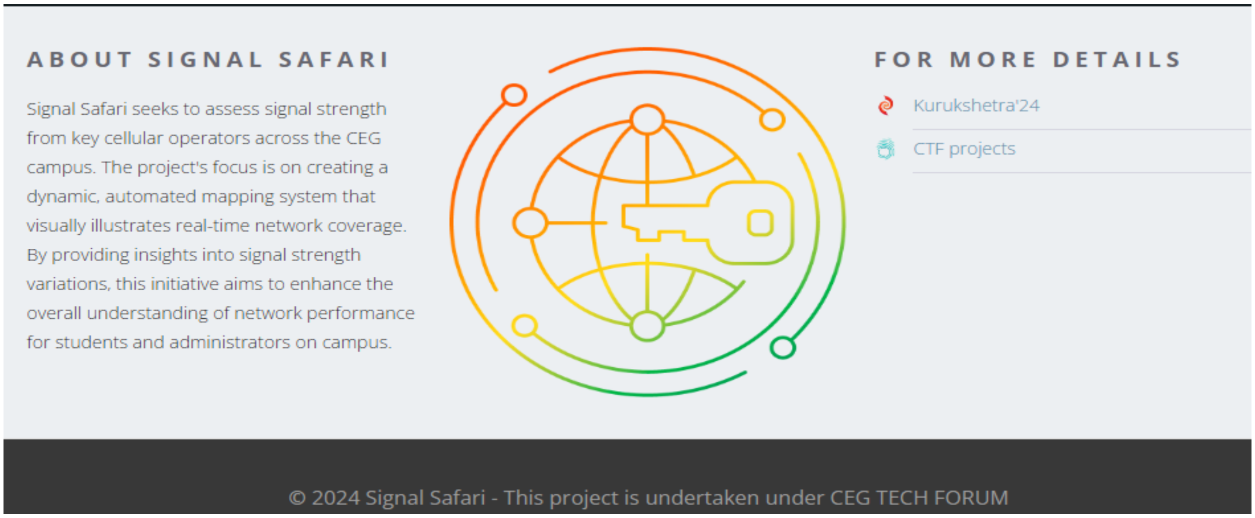




```
<div class="wrapper">
  <div class="half">
    <h4>Download speed</h4>
    <div id="map1" class="map1"></div>
    <div>
      <label>
        Speed
        <span id="info1">-</span> kbps
      </label>
    </div>
  </div>
  <div class="half">
    <h4>Upload speed</h4>
    <div id="map2" class="map1"></div>
    <div>
      <label>
        Speed
        <span id="info2">-</span> kbps
      </label>
    </div>
  </div>
</div>
```

Fig 7.3 MAP AND HTML CODE

- The other sections give a general description of the features present in the map including a footer giving external links to the Kurukshetra project site.
  - It includes a paragraph element and an anchor tag for hyperlinks
  - An image element <img src = " "> within which the link to the image is mentioned.
  - The features have been colour coded based on the colour palette used in CSS




```html
<div class="footer">
    <div class="w-container">
        <div class="w-row">
            <div class="image w-col w-col-4">
                <h5><strong>about signal safari</strong></h5>
                <p class="paragraph">Signal Safari seeks to assess signal strength from key cellular operators ...
                    performance for students and administrators on campus.</p>
            </div>
            <div class="image w-col w-col-4"><img
                    src="https://assets-global.website-files.com/65c6eeaa3c02c2b945d8b15a/65c70e1e55c43bd89cbc2d9e_signal-safari.png"
                    loading="lazy" alt="" /></div>
            <div class="w-col w-col-4">
                <h5><strong>for more details</strong></h5>
                <div class="footer-link-wrapper w-clearfix"><img ...
                        width="20" alt="" class="info-icon" /><a href="https://kurukshetraceg.org.in/ctfprojects"
                        target="_blank" class="footer-link with-icon">Kurukshetra&#x27;24</a></div>
                <div class="footer-link-wrapper w-clearfix"><img ...
                        alt="" class="info-icon" /><a href="https://cegtechforum.in/" target="_blank"
                        class="footer-link with-icon">Ceg Tech Forum</a></div>
            </div>
        </div>
    </div>
</div>
<div class="footer center">
    <div class="w-container">
        <div class="footer-text"><strong>©</strong> 2024 Signal Safari - This project is undertaken under CEG TECH
            FORUM</div>
    </div>
</div>
```

Fig 7.4WEBPAGE AND HTML CODE - 2

- A proper ALT attribute has been added to the images for cases of poor connection ideal for screen readers
- The website has been made compatible with all devices including mobile phones, tablets, laptops, and desktop machines.
- The code has been run with three files each for HTML, CSS, and Javascript (index.html, app.css, and script.js)

Testing

- The first stage prototype website was created and later tested on various devices to check its compatibility and the working of alt elements.
- The lines of code went through several debugging stages to enhance the visual effects.

7.5 REVIEW AND FEEDBACK

- After the first stage of development and testing, it was time for review and feedback from team members. The feedback included the following changes to be done:
    - All the sections have to be fitted into the frame
    - The static map has to be replaced by the dynamic map
    - The map must have a hover effect to find the intermediate signal strength values across campus.
    - Link the footer CTF link to projects.cegtechforum.in
    - On clicking a point on the map, its upload and download speed must pop up.
    - The map section must be rearranged to bring it after the homepage.

- All these changes were made, changing the static to a dynamic map which looks something like this at an instant.

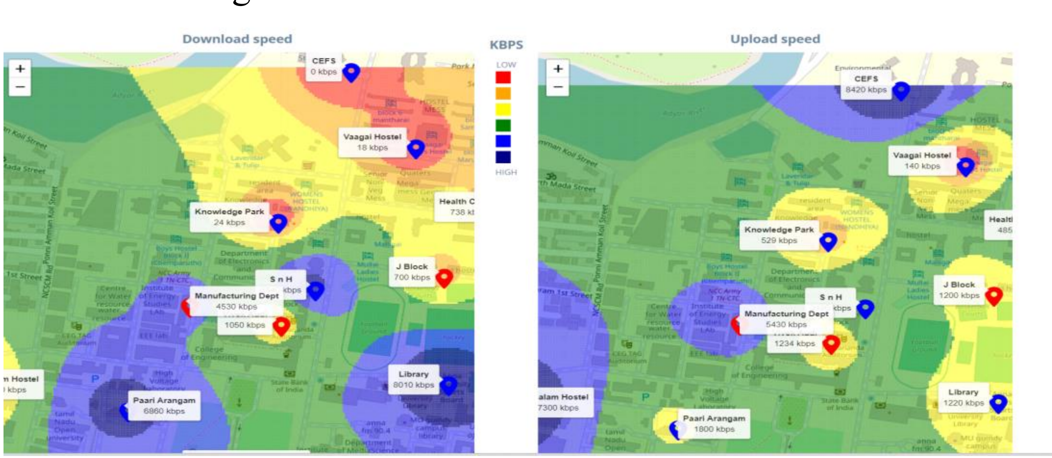


Fig 7.5 DYNAMIC MAP OUTPUT

7.6 DEPLOYMENT

- The final step of the process involves deploying the web project to any cloud platform. One such platform we used is Versel.
- Vercel is a cloud platform for static sites and serverless functions. It allows developers to build and deploy web projects with ease
- It also has a user-friendly interface and integrations with popular development tools like GitHub.
- Vercel allows for automatic deployments on every branch push and merges onto the Production Branch of your GitHub.
- This has been used to deploy the website and can be freely viewed through the link https://signal-safari.vercel.app/frontend.html

CHAPTER 8

CONCLUSION

Signal Safari embarks on a comprehensive expedition to meticulously gauge the signal integrity emanating from the principal cellular operator, Airtel, spanning the vast terrain of the CEG campus. Our ambitious project is anchored on the creation of a dynamic, automated cartographic framework that intricately maps out the ever-fluctuating network coverage in real-time. By harnessing cutting-edge technology, we aspire to provide a visually immersive experience that illuminates the nuanced variations in signal strength, thus fostering a profound understanding of network performance for the diverse community of students and administrators traversing the campus landscape. Through this endeavour, we endeavour to empower stakeholders with actionable insights, enabling informed decision-making and enhancing the overall connectivity experience within the campus ecosystem.

CHAPTER 9

REFERENCES

1. Merve Keskin et al (2015), “Comparing Spatial Interpolation Methods for Mapping Meteorological Data in Turkey”, Energy Systems and Management, Springer Proceedings in Energy, DOI 10.1007/978-3-319-16024-5_3
2. Xu Zhong et al, “Spatial Interpolation of Streaming Geosensor Network Data in the RISER system”
3. Naser Tarhuni et al (2023), “Machine-Learning-Based Ground-Level Mobile Network Coverage Prediction Using UAV Measurements”, J. Sens. Actuator Netw. 2023, 12, 44, DOI 10.3390/jsan12030044
4. https://www.youtube.com/live/48Hsnqdw7Hk?si=ebz_evwY0U7IPHe1
5. https://youtu.be/FYUhRIwHQYs?si=Rcck8d598XRUORA2
6. https://randomnerdtutorials.com/esp32-sim800l-publish-data-to-cloud/
7. https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/#hosting-php-application-mysql
8. https://github.com/Vajravegha/VVM501_ESP32_4GLTE_MODULE
9. https://randomnerdtutorials.com/esp32-cloud-mqtt-broker-sim800l/