ONBOARD SATELLITE IMAGE PROCESSING USING AI FOR ENVIRONMENTAL SUSTAINABILITY

Under the Supervision of: Associate Prof. Mohamed Mourad Abdelrahman

PRESENTED BY

- Hazem Adel Abdelrazek Ahmed
- Mohammed Abdelhakim Mohammed El-Bakry
- Areeg Tarek Mahmoud Abdullah
- Sarah Ayman Ahmed Mohammed
- Nada Hesham Abdelgaber Ali
- Mennatullah Muhammad Mahmoud Ali
- Radwa Omar Ali Mohammed
- Menna Mohamed Fathy Ali
- Adel Ibrahim Ibrahim Hashem
- Manar Mohamed Allam Soliman
- Mohammed Abdelfattah Mohammed Ahmed
- Ahmed Abdelradi Ahmed Fouad

The life of a
FISHERMAN

A journey of patience, strength, and the sea.

WHY FISHING

- Food security threat
- Production collapse
- Livelihood emergency
- Aquaculture imbalance

RED SEA

- Biodiversity jewel
- Shocking neglect
- Perfect for satellites
- Closed sea


Catch More Fish !

INTRODUCTION

ARTIFICIAL INTELLIGENCE IN ENVIRONMENTAL MONITORING

In this project, Artificial Intelligence plays a key role in analyzing satellite imagery to support environmental sustainability — specifically by predicting vital oceanographic variables and detecting potential fish aggregation zones in the Red Sea.

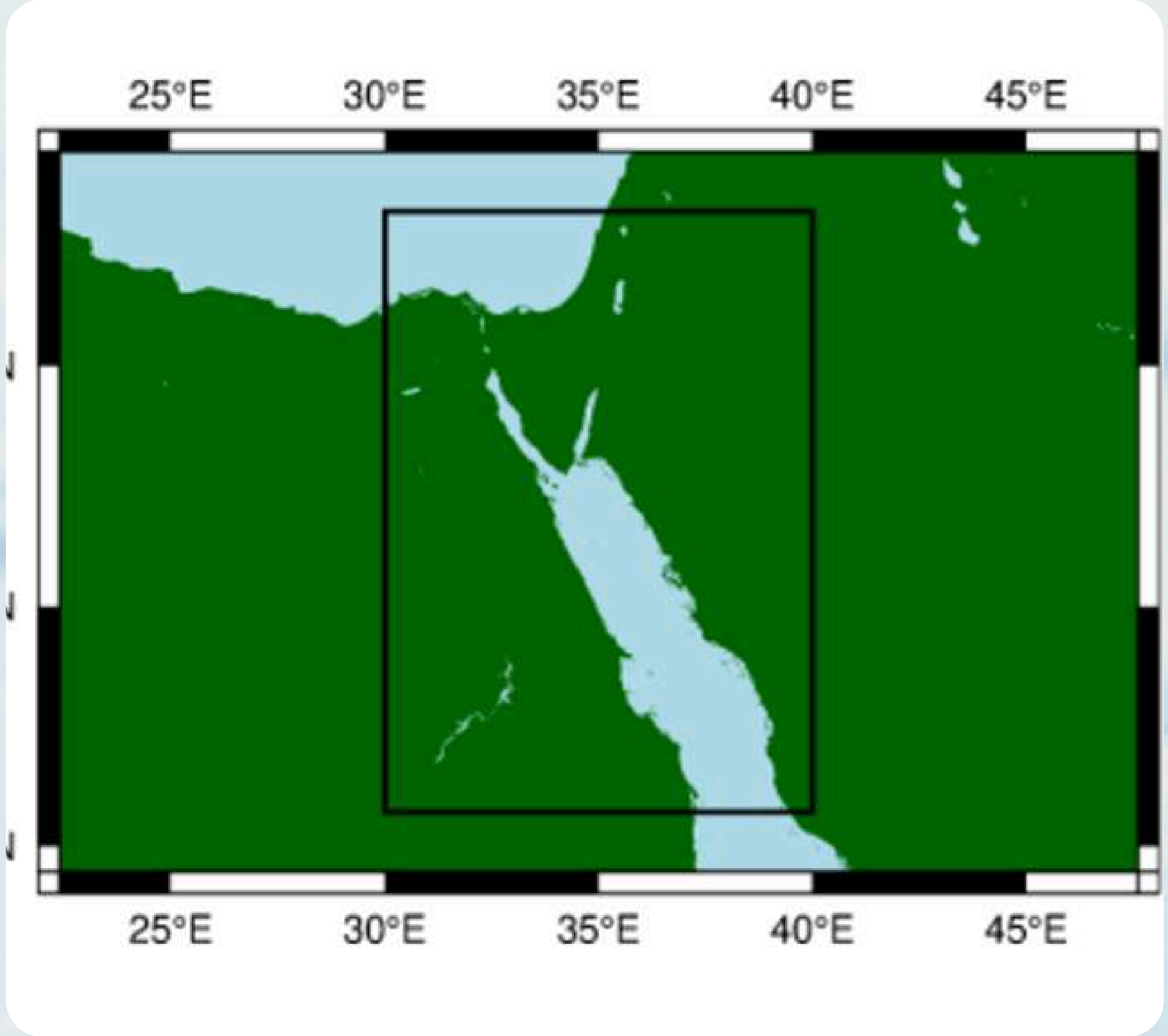
WHY AI?

- Traditional image processing can't handle the volume, complexity, or variability of satellite data effectively.
- AI allows for:
  - Learning complex patterns in spatiotemporal data
  - Predicting future environmental behavior
  - Automating detection of fish zones based on biological preferences

---

OVERALL GOAL

To transform raw satellite data into actionable insights that guide sustainable fishing — using accurate, automated, and interpretable AI models.



Study Area – Red Sea, Egypt

Geographic Focus:

- Region: Northern Red Sea – Egyptian waters
- Latitude Range: 24.5°N to 26.5°N
- Longitude Range: 33.5°E to 35.5°E

Environmental Factors Affecting Fish Distribution

01. Sea surface Temperature (SST)

02. Chlorophyll-a

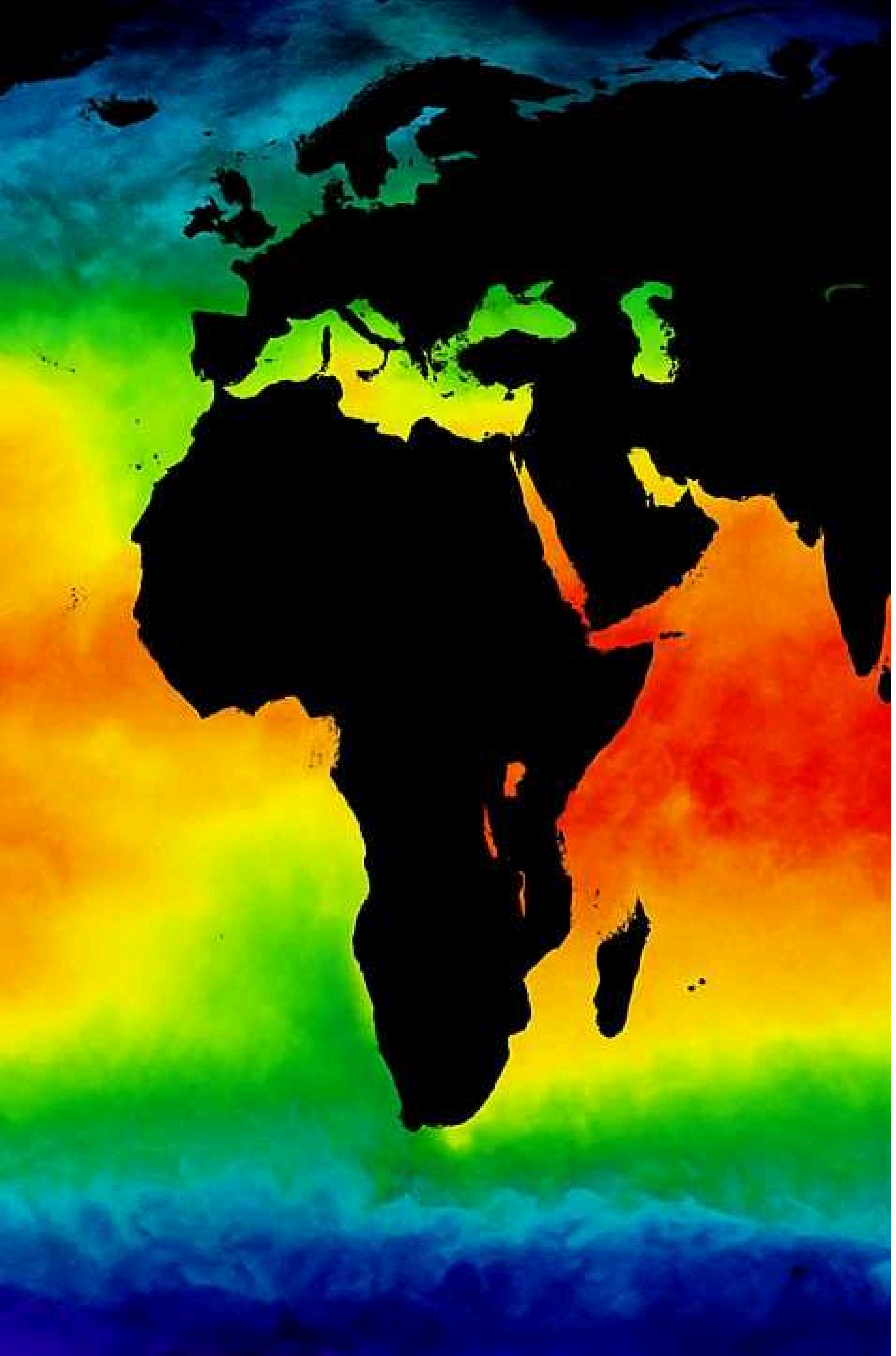
03. Salinity

04. Dissolved Oxygen (DO)

05. Wind and Currents

06. Habitat Structure

Why Chlorophyll-A And SST Were Chosen ?



they serve as key drivers of marine biological activity.

Numerous studies have demonstrated their direct correlation with fish abundance and migration patterns,

making them highly reliable indicators for identifying potential fishing grounds.

DATA ACQUISITION & PREPROCESSING

The chlorophyll-a concentration data were obtained from the Moderate Resolution Imaging Spectroradiometer (MODIS) onboard NASA's Aqua satellite, with a spatial resolution of approximately 1 km. Likewise, SST data were acquired from the same sensor to ensure consistency in spatial and temporal resolution

1-**Data Source:**

🌐 All datasets were sourced from the NASA Ocean Color Web

2-Preprocessing Highlights

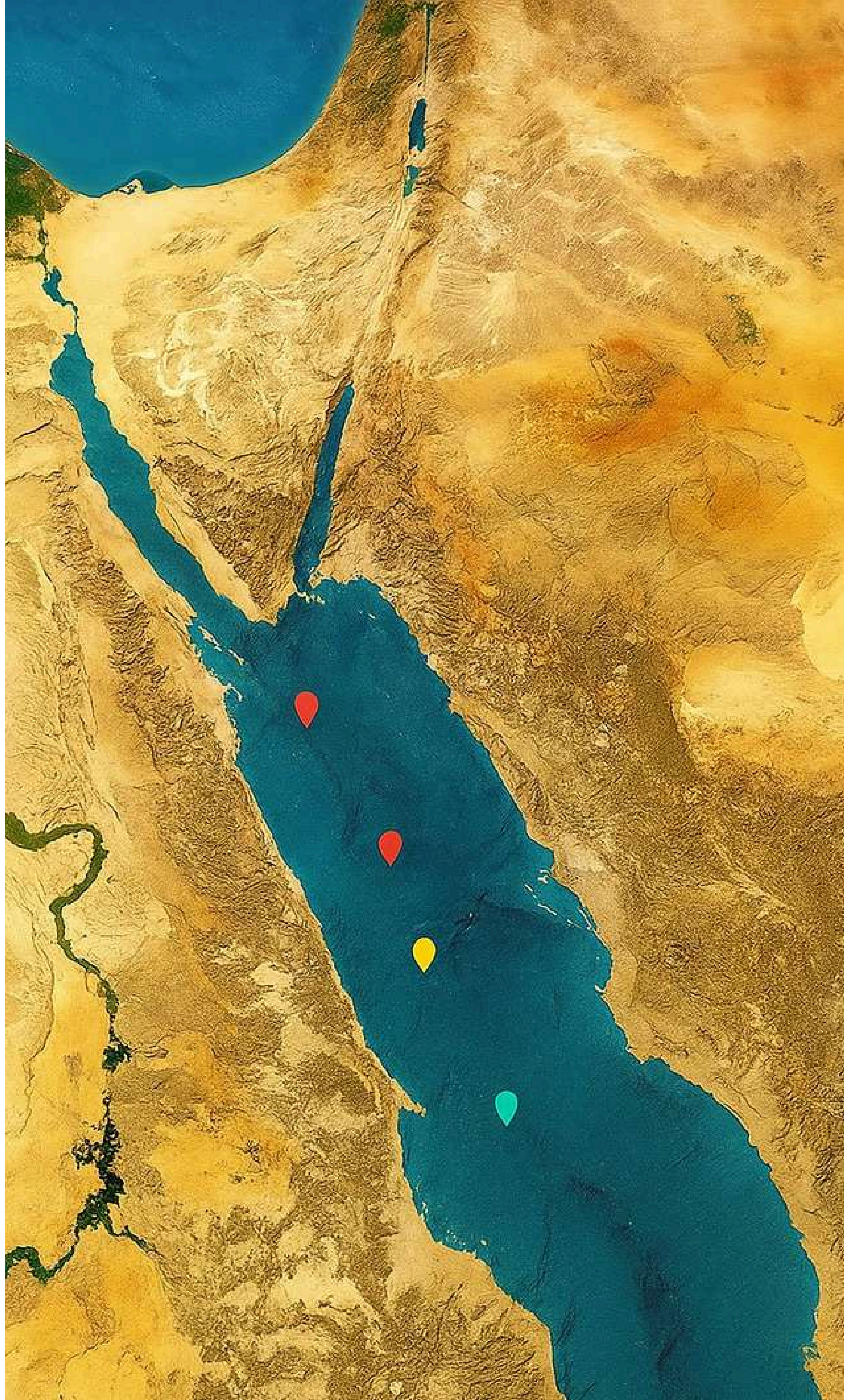
Model Design

1-Detection Algorithm for Identifying Potential Fish Presence Zones

2-Prediction Models

Detection

Fish detection refers to the process of identifying and locating the presence of fish in marine environments using remote sensing data.

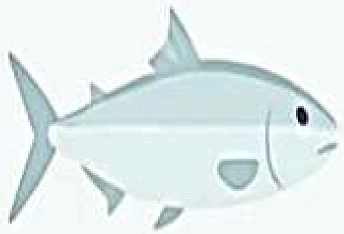
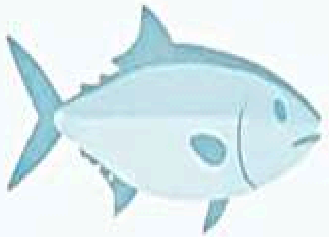


Types of fish and their presence

Detection Algorithm

We developed a simple rule-based algorithm that uses satellite data — mainly sea surface temperature and chlorophyll levels — to identify areas in the sea where fish are most likely to be found. These two factors play a major role in where fish live, move, and gather.

Preferred Environmental Conditions for Some Marine Species

Species	SST Range (°C)	Chlorophyll-a (mg/m^3)	Ecological Interpretation
 Sardine	18 - 24	0,3 - 1.0	Prefers moderate temperatores and high phytoplankton abundance
 Tuna	>24	0,2 - 0.6	Associated with warm waters and moderate productivity
Mackerel	20 - 26	0,4 - 0.9	Found in moderately warm, nutrient-rich areas
Shrimp	22 - 28	0,5 - 1.2	

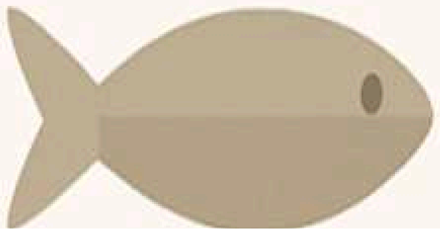
Fish quantity equation

We designed a scoring system that estimates how suitable the current sea conditions are for each fish type.

The algorithm checks if the temperature and chlorophyll values fall within the preferred range for the fish.

If yes, it calculates how close those values are to the optimal center of that range — the closer they are, the higher the score.

```
for  fish, t_min, t_max, c_max, max_qty in fish_
    if t_min <= temp <= t_max and c_chl <= c_max
    # Flexible abundance index by proximity to
    the middle
    t_center = (t_min + t_max) / 2
    c_center = (c_min + c_max) / 2
    t_score = 1 - abs(temp - t_center) /
    ((t_max - chl) / (((c_max - c_min) / 2)
    score = max(0, (t_score + c_score) / 2)
    estimated_qty = int(score * max_qty)
```



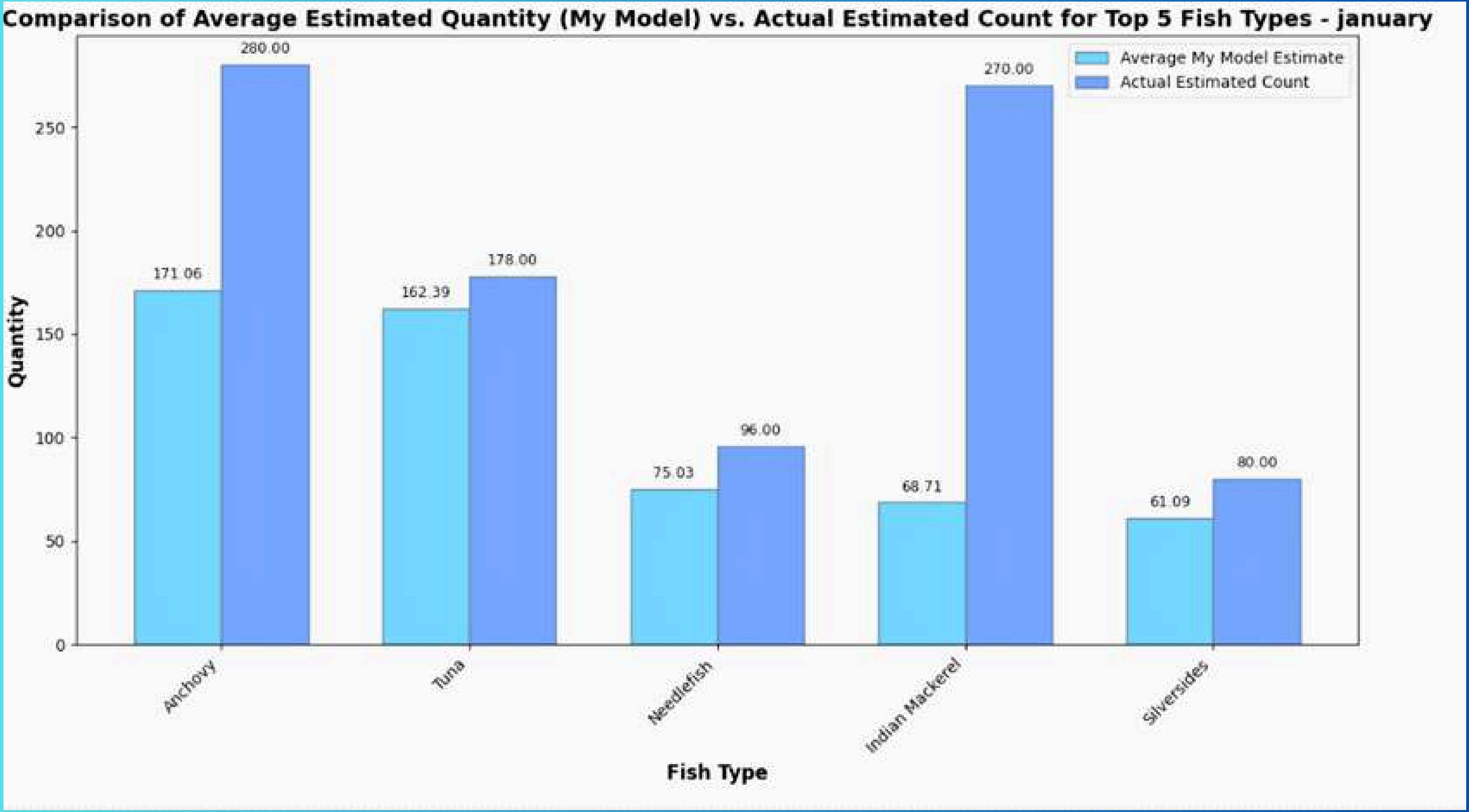
Then it estimates the potential fish quantity based on that score

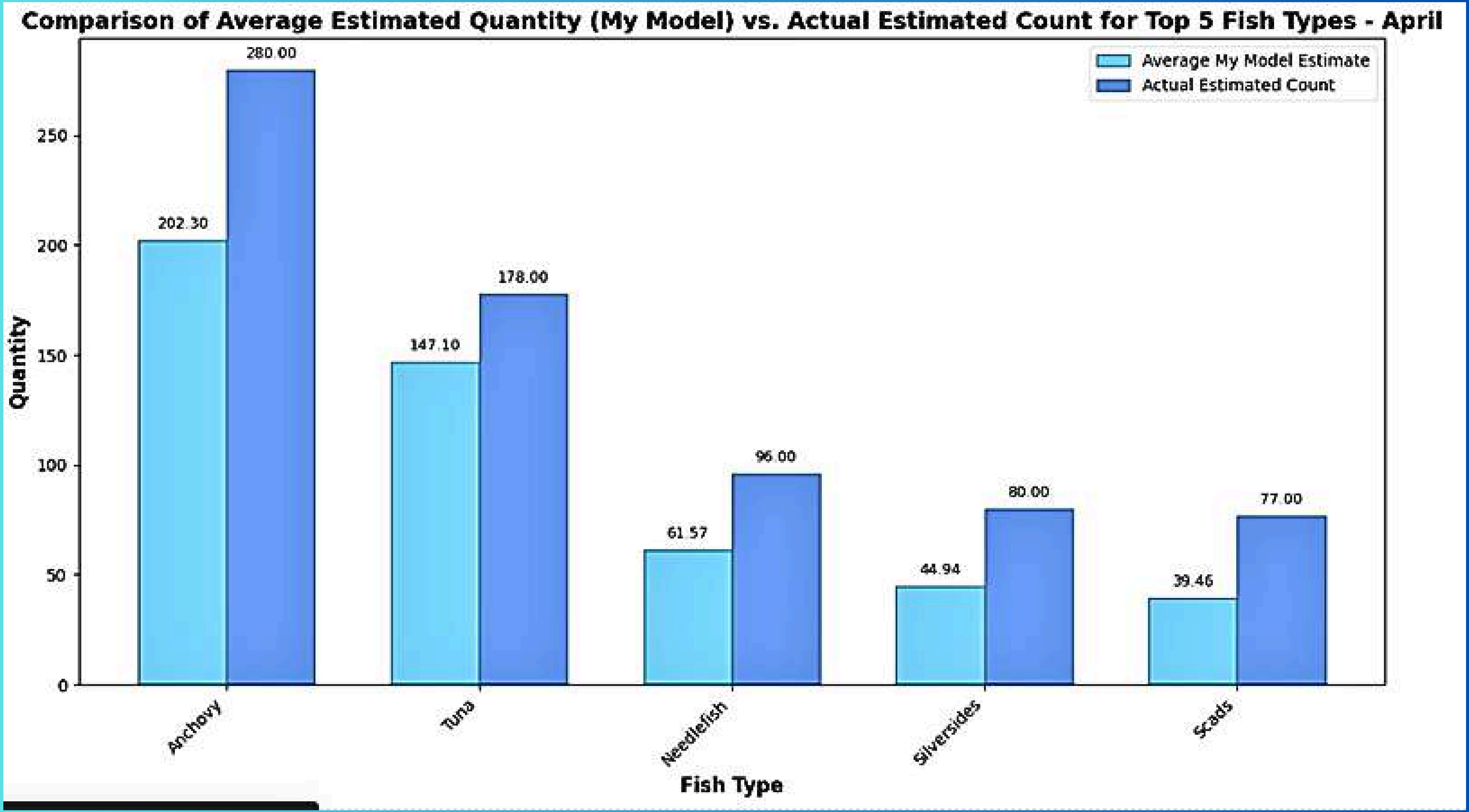
For example, if tuna prefers temperatures between 24–30°C and chlorophyll between 0.1–0.3, and today's values are 27°C and 0.2 — that's near perfect.

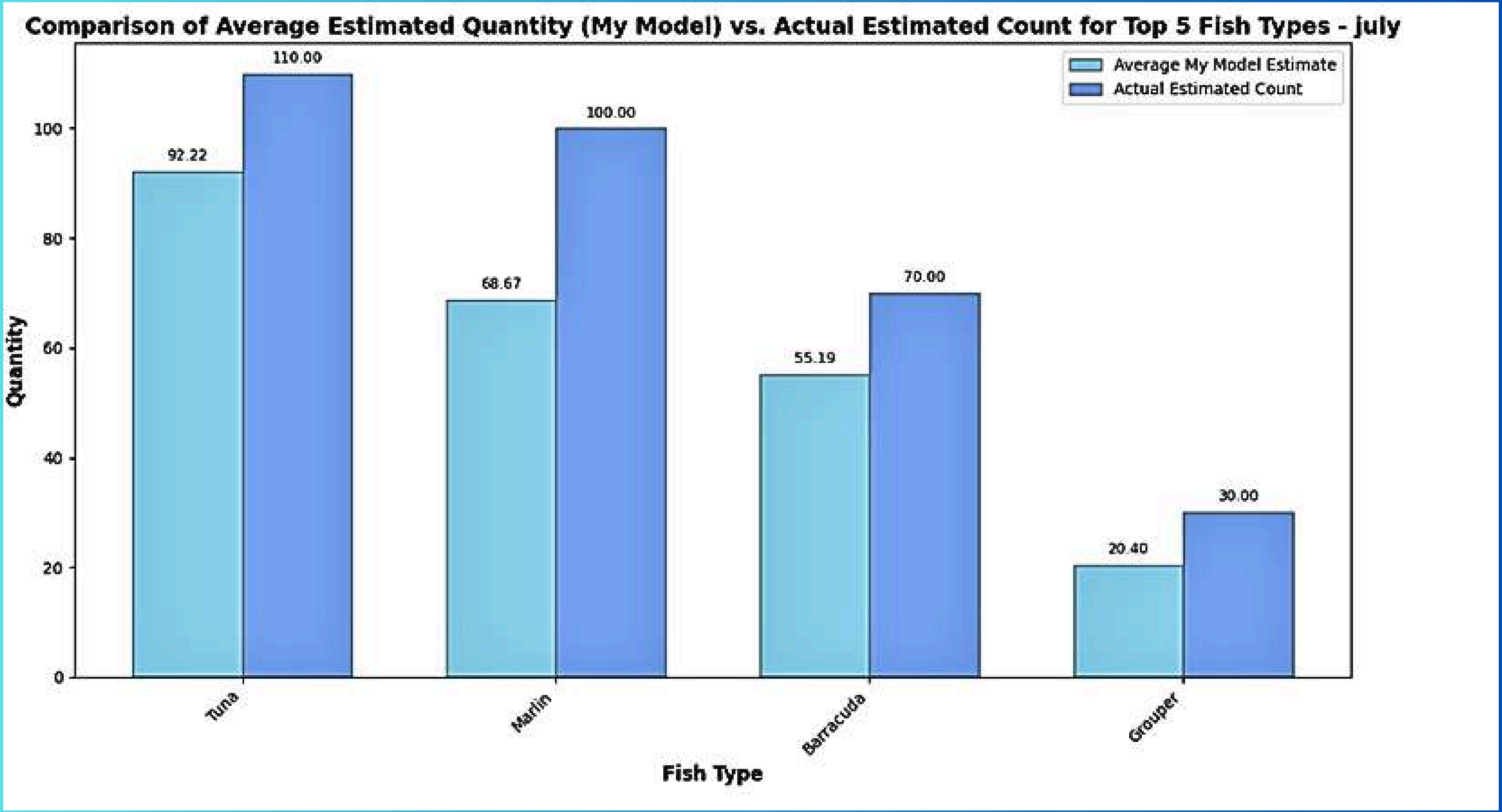
So the algorithm gives it a high score and estimates high fish abundance in that area.

	fish_type	Actual Estimated Count
0	Tuna	178
1	Sardine	3600
2	Mackerel	1380
3	Shrimp	6800
4	Baga	630
5	Mullets	1100
6	Anchovy	2400
7	Grouper	340
8	Sea Bream	620
9	Barracuda	160
10	Snapper	430
11	Trevally	520
12	Rabbitfish	620
13	Emperor	410
14	Jackfish	480

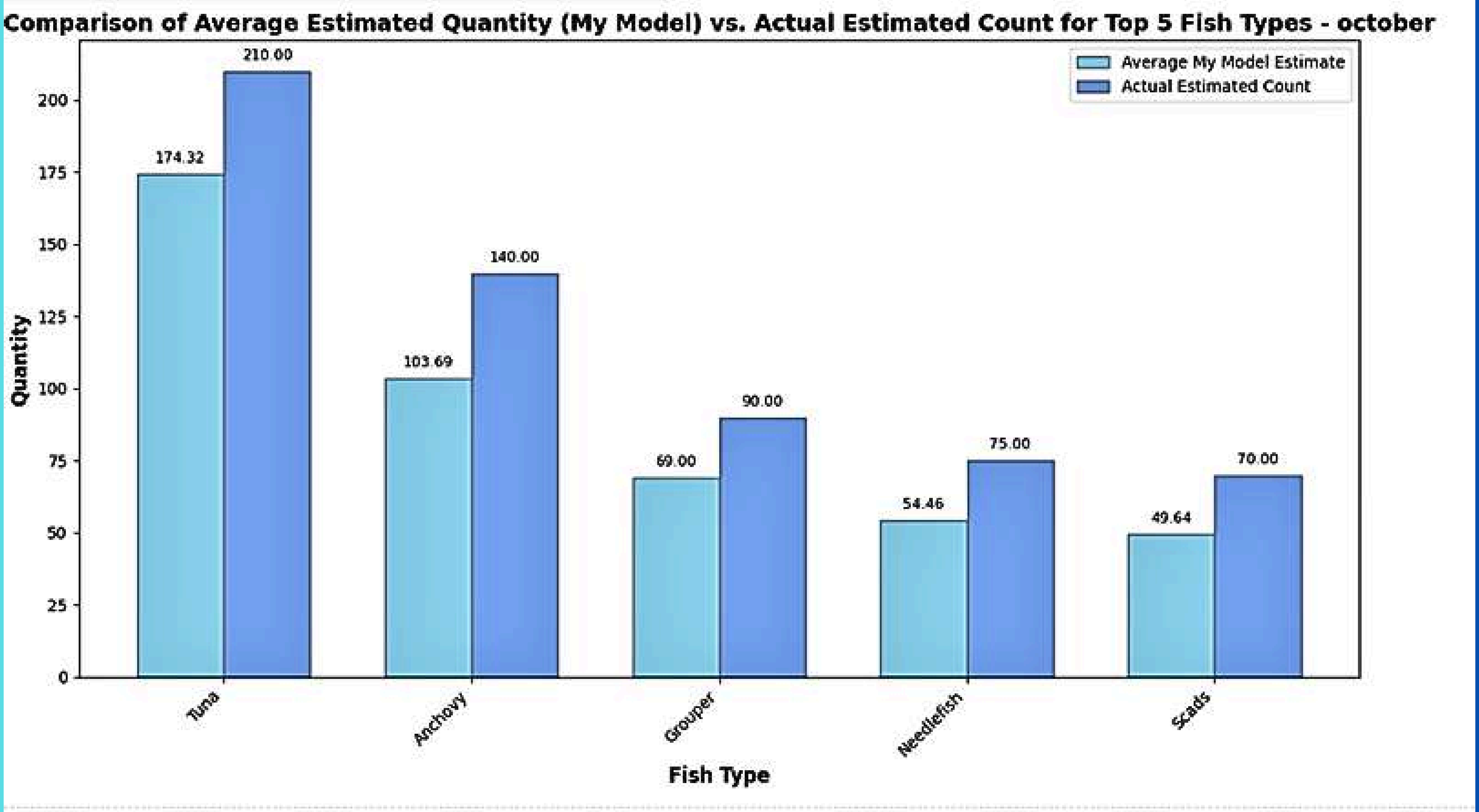
Seasonal Analysis Of Fish Detection Algorithm Results Compared To Actual Observations


Comparison of Average Estimated Quantity (My Model) vs. Actual Estimated Count for Top 5 Fish Types - january
Average My Model Estimate
Actual Estimated Count
Quantity
Fish Type
Anchovy
Tuna
Needlefish
Indian Mackerel
Silversides
171.06
280.00
162.39
178.00
75.03
96.00
68.71
270.00
61.09
80.00
0
50
100
150
200
250


Comparison of Average Estimated Quantity (My Model) vs. Actual Estimated Count for Top 5 Fish Types - April
Average My Model Estimate
Actual Estimated Count
Quantity
0
50
100
150
200
250
202.30
280.00
147.10
178.00
61.57
96.00
44.94
80.00
39.46
77.00
Anchovy
Tuna
Needlefish
Silversides
Scads
Fish Type



Comparison of Average Estimated Quantity (My Model) vs. Actual Estimated Count for Top 5 Fish Types - July
Average My Model Estimate
Actual Estimated Count
Quantity
0
20
40
60
80
100
92.22
110.00
68.67
100.00
55.19
70.00
20.40
30.00
Tuna
Marlin
Barracuda
Grouper
Fish Type



Comparison of Average Estimated Quantity (My Model) vs. Actual Estimated Count for Top 5 Fish Types - october
Average My Model Estimate
Actual Estimated Count
Quantity
Fish Type
0
25
50
75
100
125
150
175
200
174.32
210.00
103.69
140.00
69.00
90.00
54.46
75.00
49.64
70.00
Tuna
Anchovy
Grouper
Needlefish
Scads

2-

PREDICTION MODELS

How Do We Measure Model Performance?

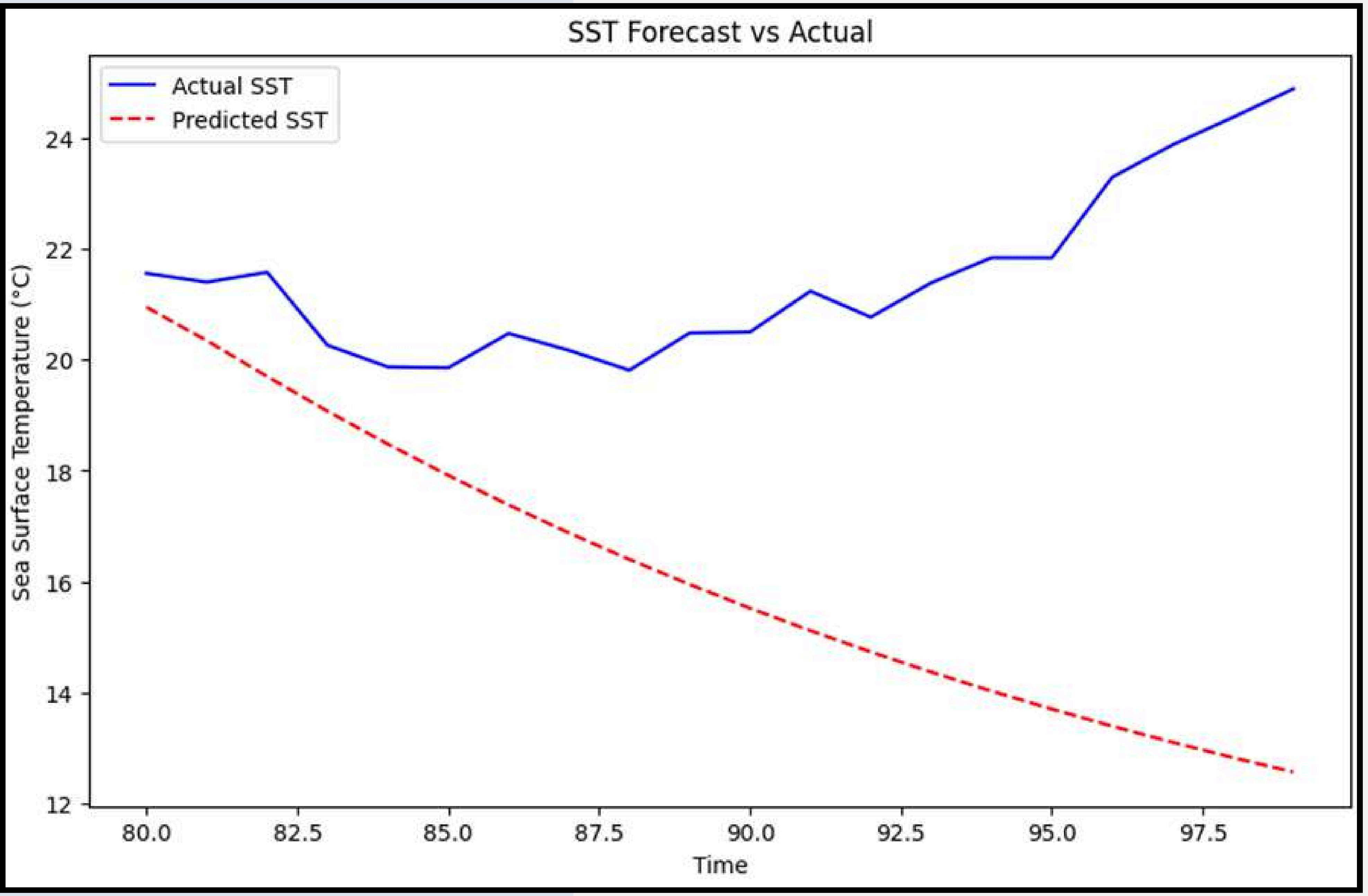
We used standard regression evaluation metrics to assess prediction accuracy:

Key Metrics:

1. R^2 Score (Coefficient of Determination):
   - Measures how well predictions explain the variance in actual values
   - Closer to 1.0 → Better model performance
   - Example: "R^2 = 0.97" means the model explains 97% of the variance
2. Mean Absolute Error (MAE):
   - Average of absolute differences between predicted and actual values
   - Lower MAE = more accurate
   - Easy to interpret (e.g., degrees °C or mg/m^3)
3. Mean Squared Error (MSE):
   - Penalizes larger errors more than MAE
   - Useful for detecting unstable or noisy predictions

Model Development

STAGE 1: STATISTICAL BASELINE (ARIMA)



To initiate predictive modeling, we began with a statistical baseline using the ARIMA (AutoRegressive Integrated Moving Average) model to forecast Sea Surface Temperature (SST).

Dataset Setup:

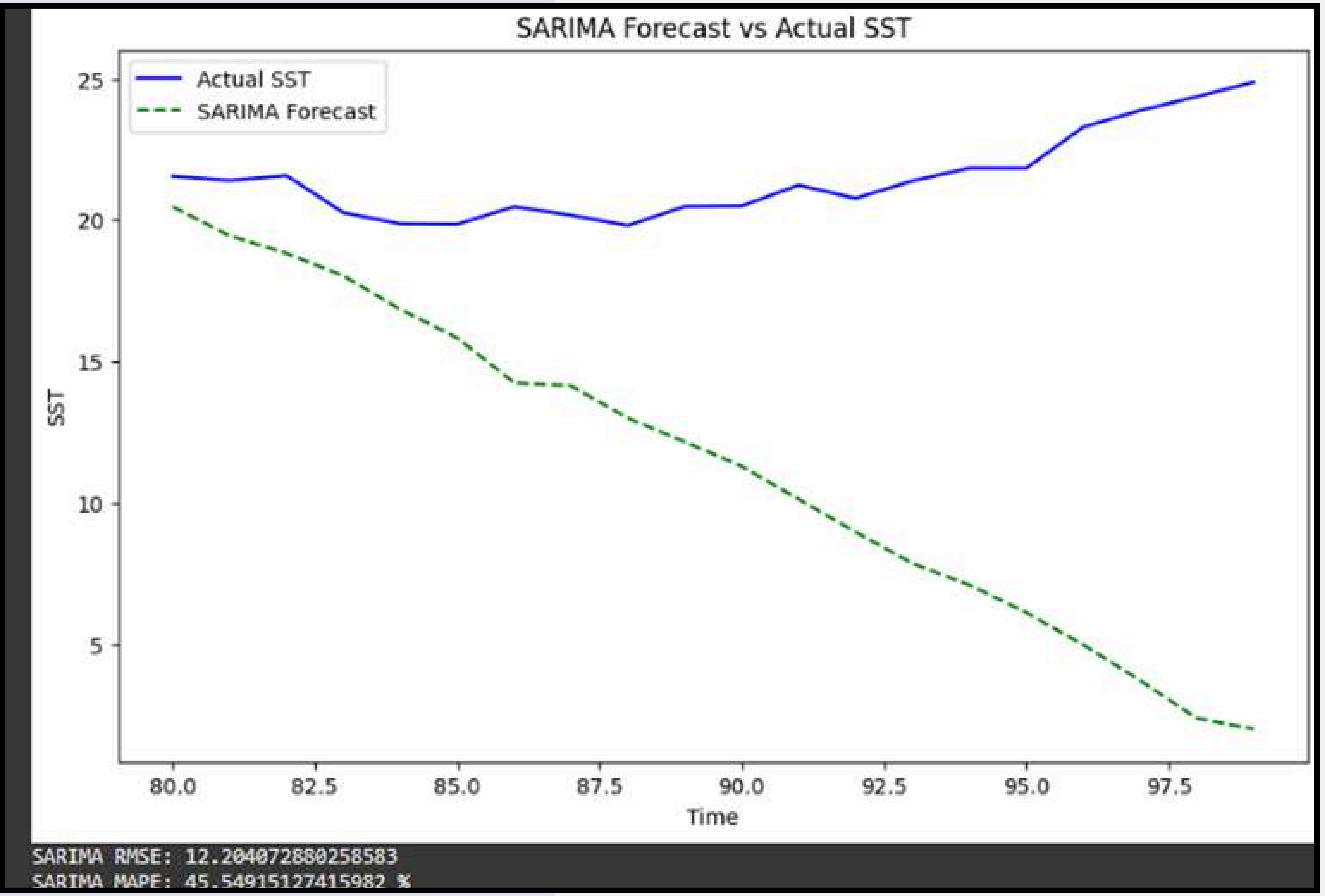
- Input: 2-year SST and Chlorophyll-a dataset
- Forecast Target: SST (univariate time-series)
- Split: 80% training, 20% testing
- ARIMA Order: (2,1,2)

Observations:

- Generated a smooth forecast curve capturing overall SST trends
- Failed to capture seasonal and nonlinear fluctuations
- Could not incorporate multivariate influences (e.g., chlorophyll-a)
- Served as a baseline reference before deep learning integration

Model Development

STAGE 2: SEASONAL FORECASTING WITH SARIMA



To better capture the seasonal behavior of SST, we upgraded from ARIMA to SARIMA (Seasonal ARIMA), trained on an expanded 4-year dataset.

Model Parameters:

- Dataset: 4 years of SST
- Train-Test Split: 80% / 20%
- ARIMA Order: (2,1,2)
- Seasonal Order: (1,1,1,12) → Models annual seasonality (12 months)

Evaluation Results:

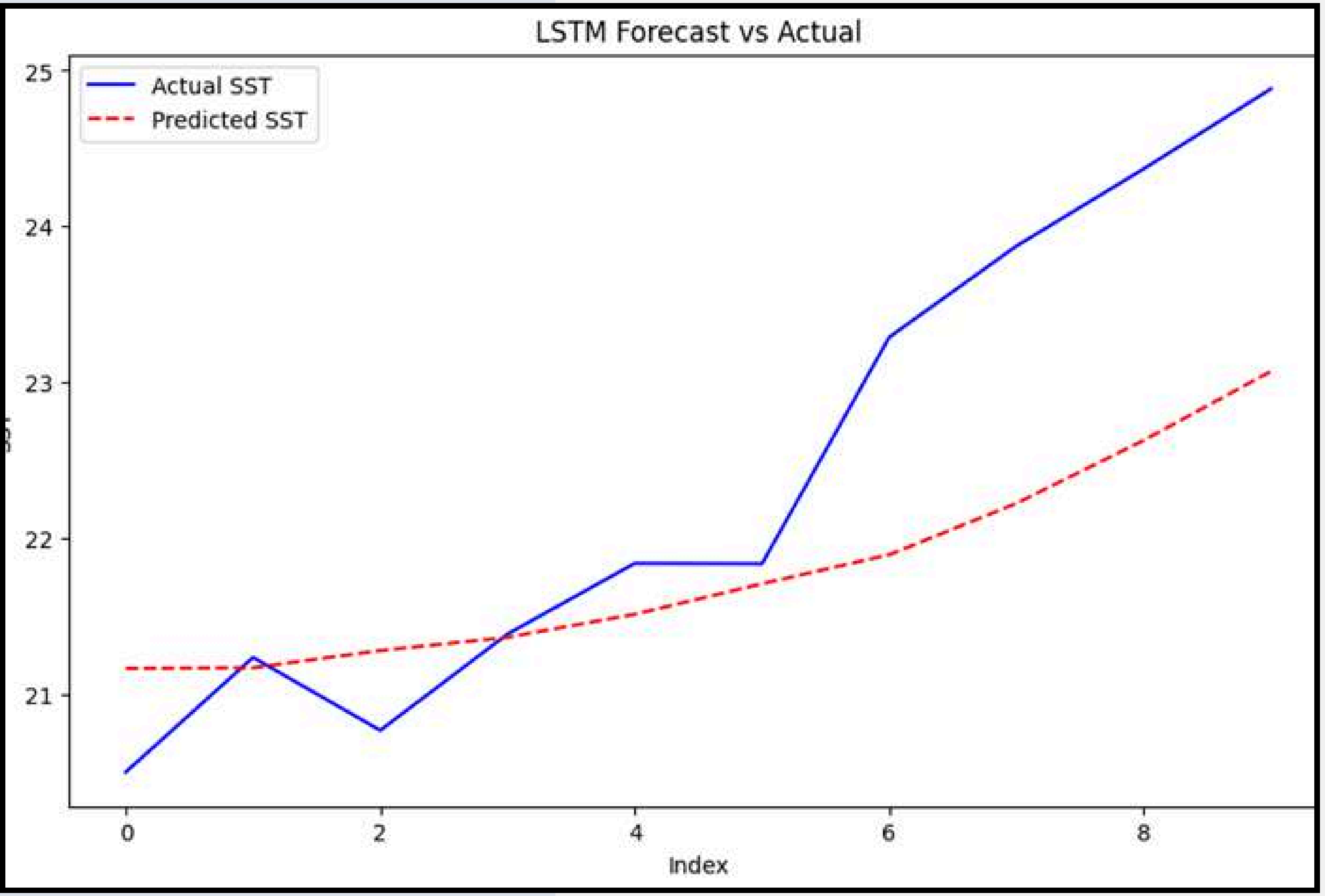
- RMSE: 12.20
- MAPE: 45.55%
- Improved seasonal trend detection over ARIMA

Key Insight:

SARIMA introduced seasonal awareness,
but its statistical nature limited flexibility and multi-feature learning.
It highlighted the need for non-linear, data-driven AI models.

Model Development

STAGE 3: DEEP LEARNING WITH LSTM



To address the limitations of statistical models, we built a Long Short-Term Memory (LSTM) neural network capable of learning nonlinear spatiotemporal patterns from satellite data.

Why LSTM?

Model Setup:

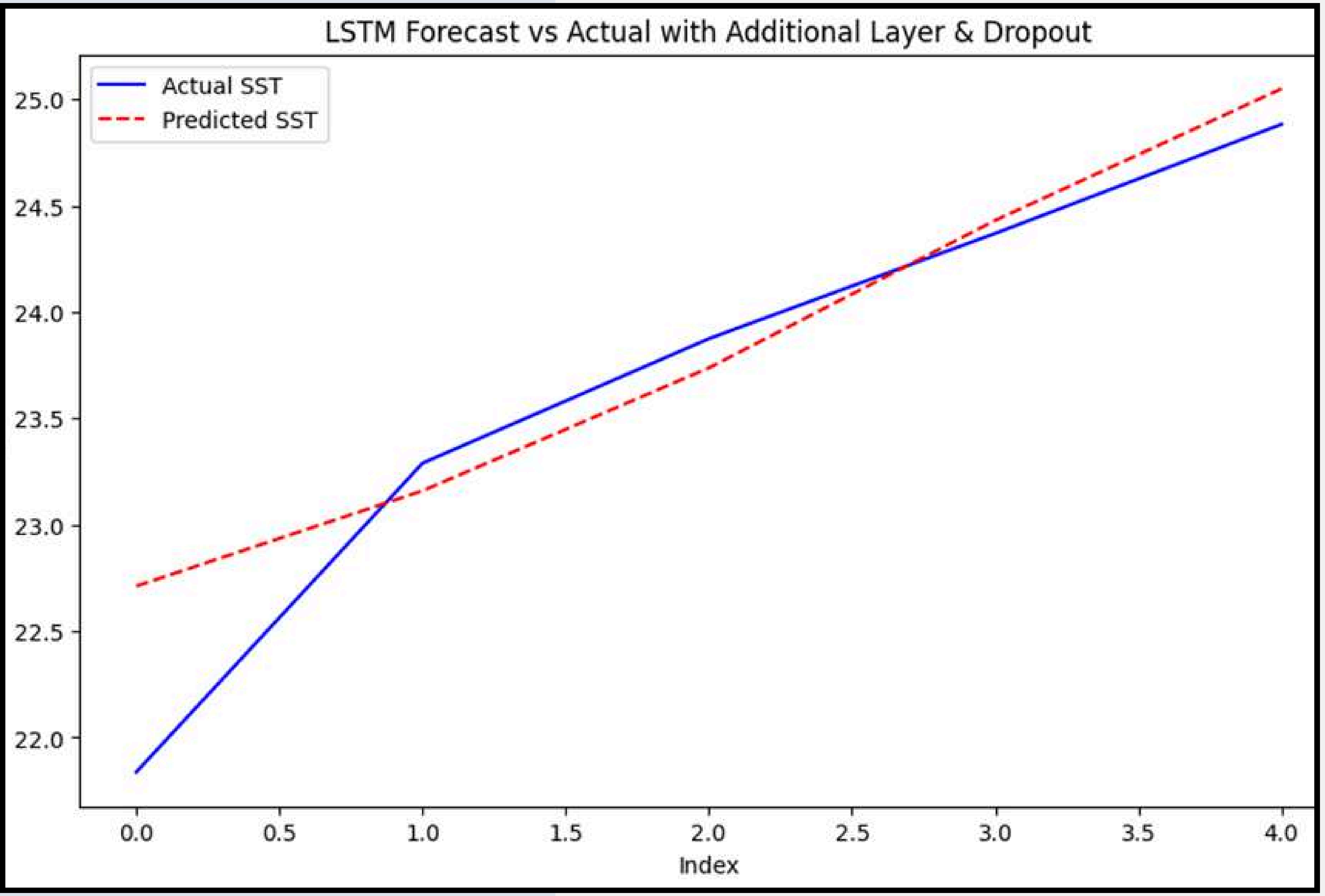
- Inputs: Chlorophyll-a, SST, Latitude, Longitude
- Window size: 12 time steps
- Architecture:
  - 1 LSTM layer (64 units, tanh)
  - 1 Dense output layer)
- Train/Test Split: 80% / 20%

Performance Metrics:

- RMSE: 1.087
- MAPE: 3.55%
- Clear improvement over ARIMA & SARIMA

Model Development

STAGE 4: ENHANCED LSTM WITH MULTIVARIATE INPUTS



Building on our earlier LSTM success, we developed an enhanced deep learning model that incorporates both SST and Chlorophyll-a, reflecting their ecological interdependence.

Performance:

- RMSE: 0.408
- MAPE: 1.21%
- Strong fit to real SST values
- Captures subtle seasonal & ecological dynamics
- High generalization with minimal overfitting

Model Development

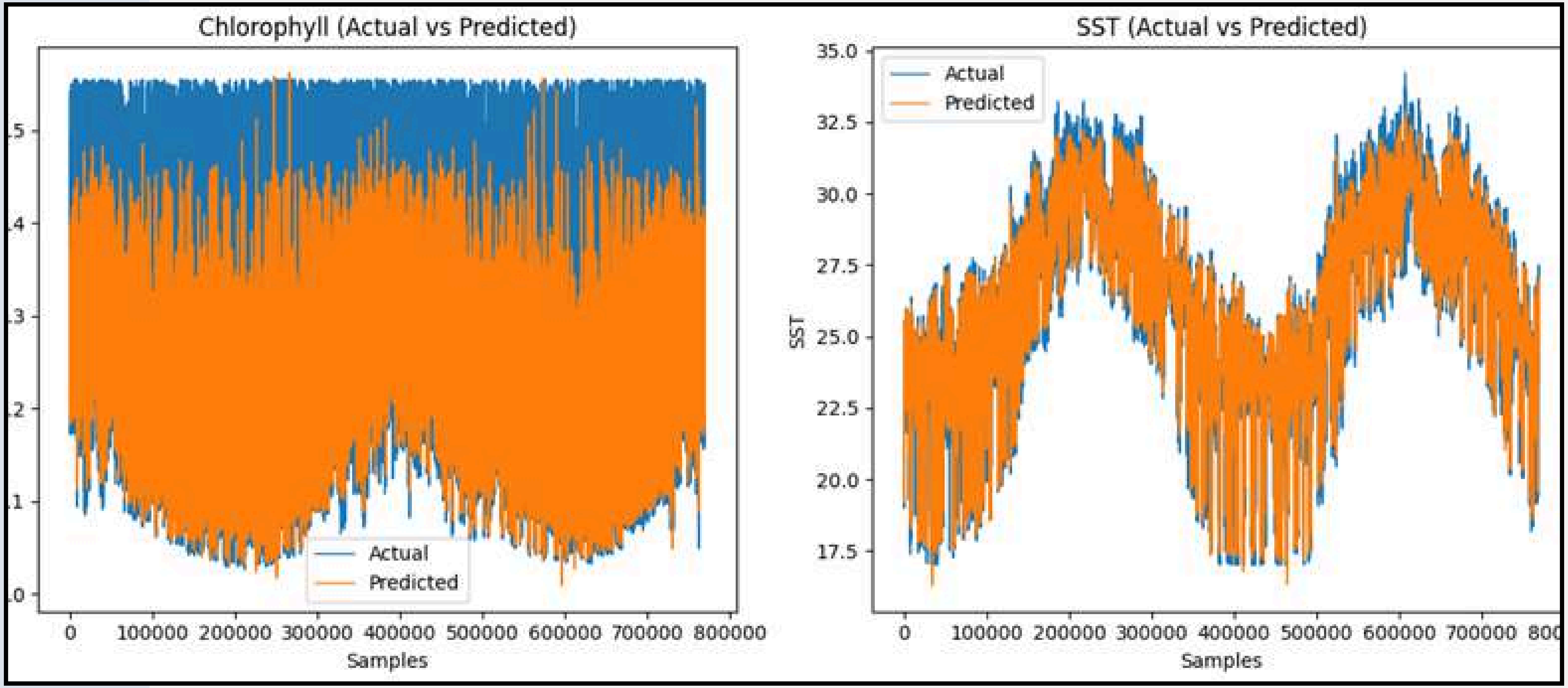
FINAL DEEP LEARNING MODEL: DUAL-OUTPUT LSTM

We designed a final LSTM-based deep learning model that predicts both Chlorophyll-a and Sea Surface Temperature (SST) simultaneously.
It represents the core of our onboard intelligent system for fish zone prediction.

Output	MSE	MAE	R^2 Score
Chlorophyll-a	0.0039	0.04	0.64
SST	0.39	0.38	0.96

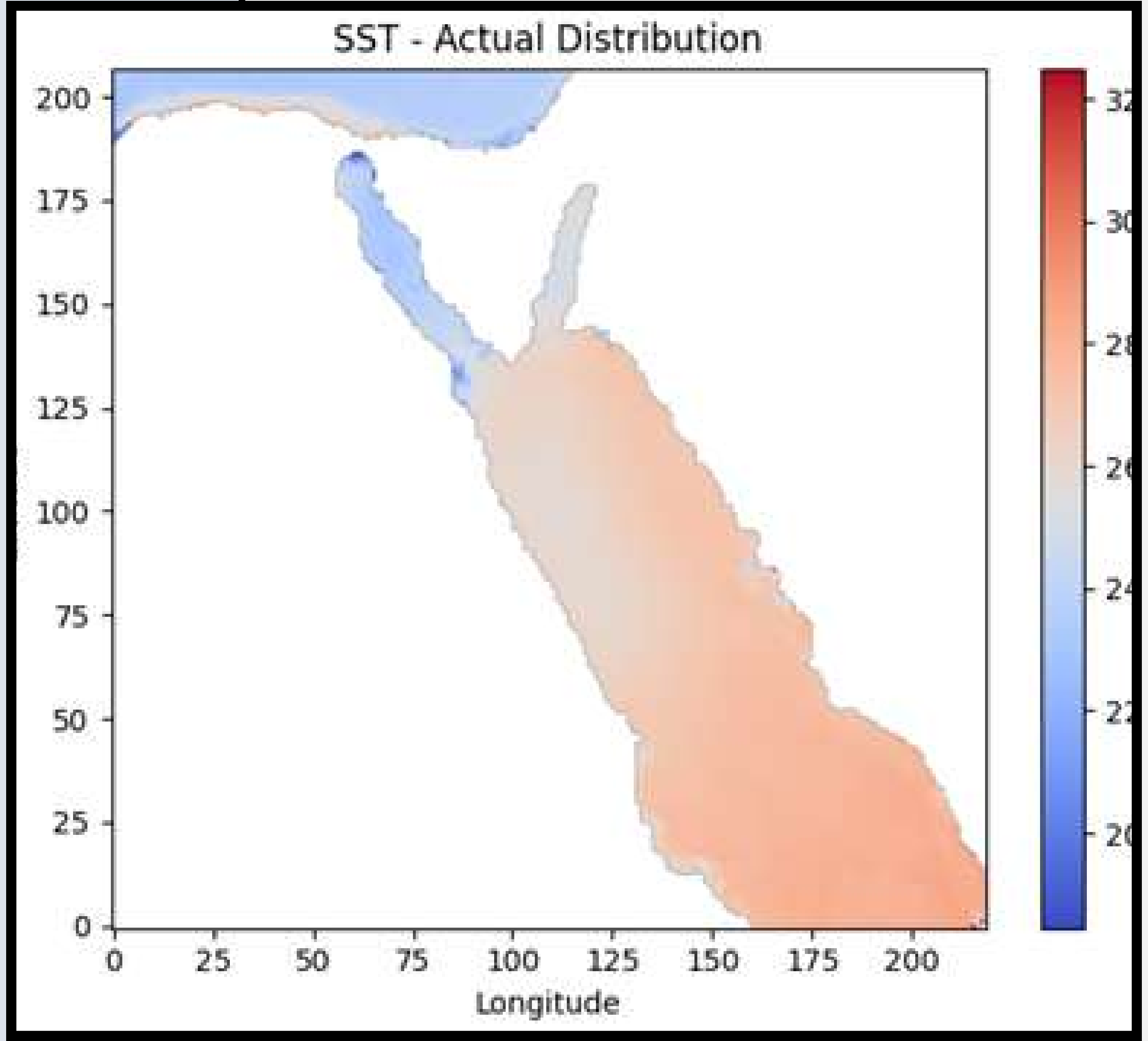
Model Development

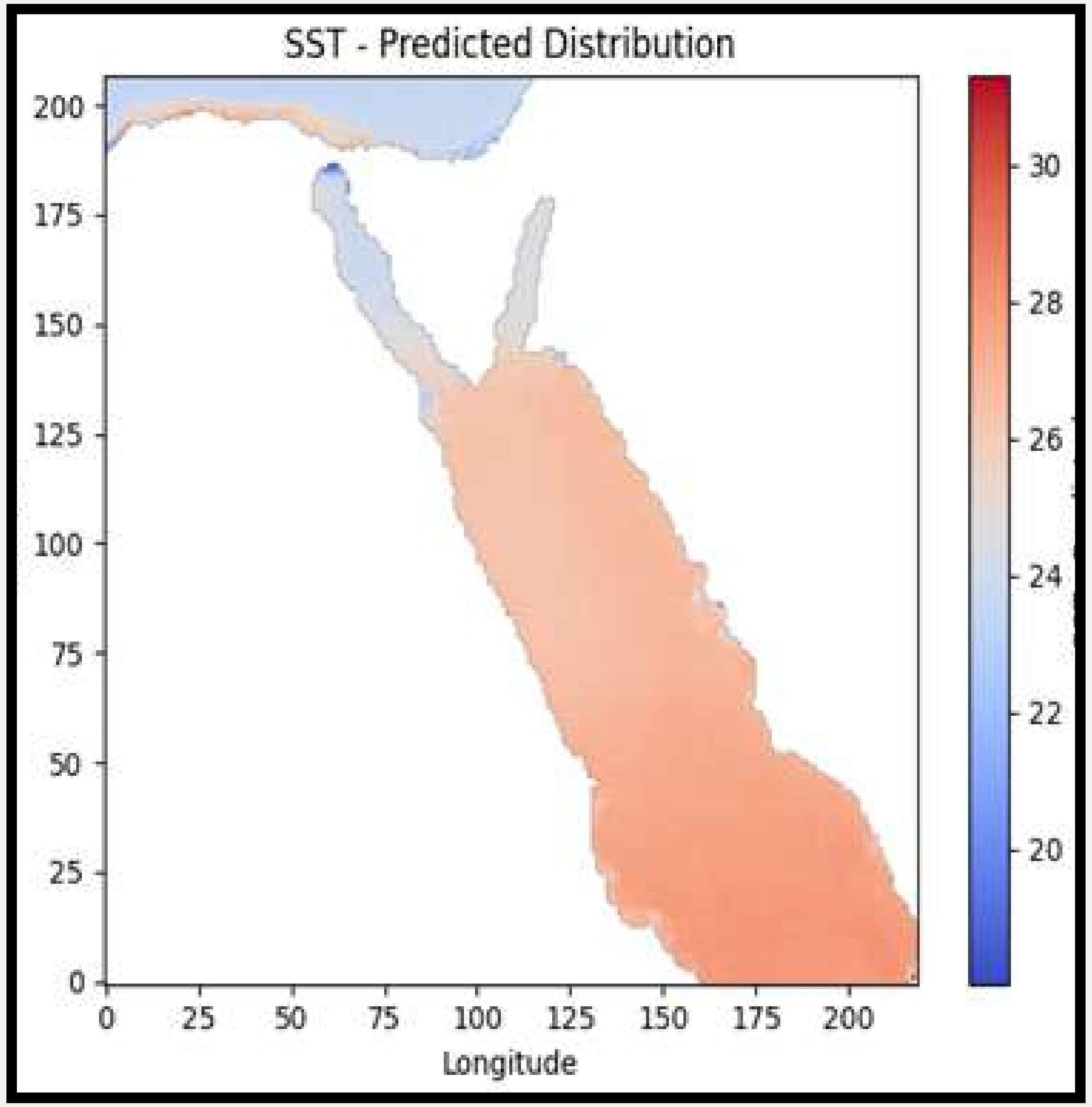
LSTM MODEL – FORECAST RESULTS & SPATIAL EVALUATION



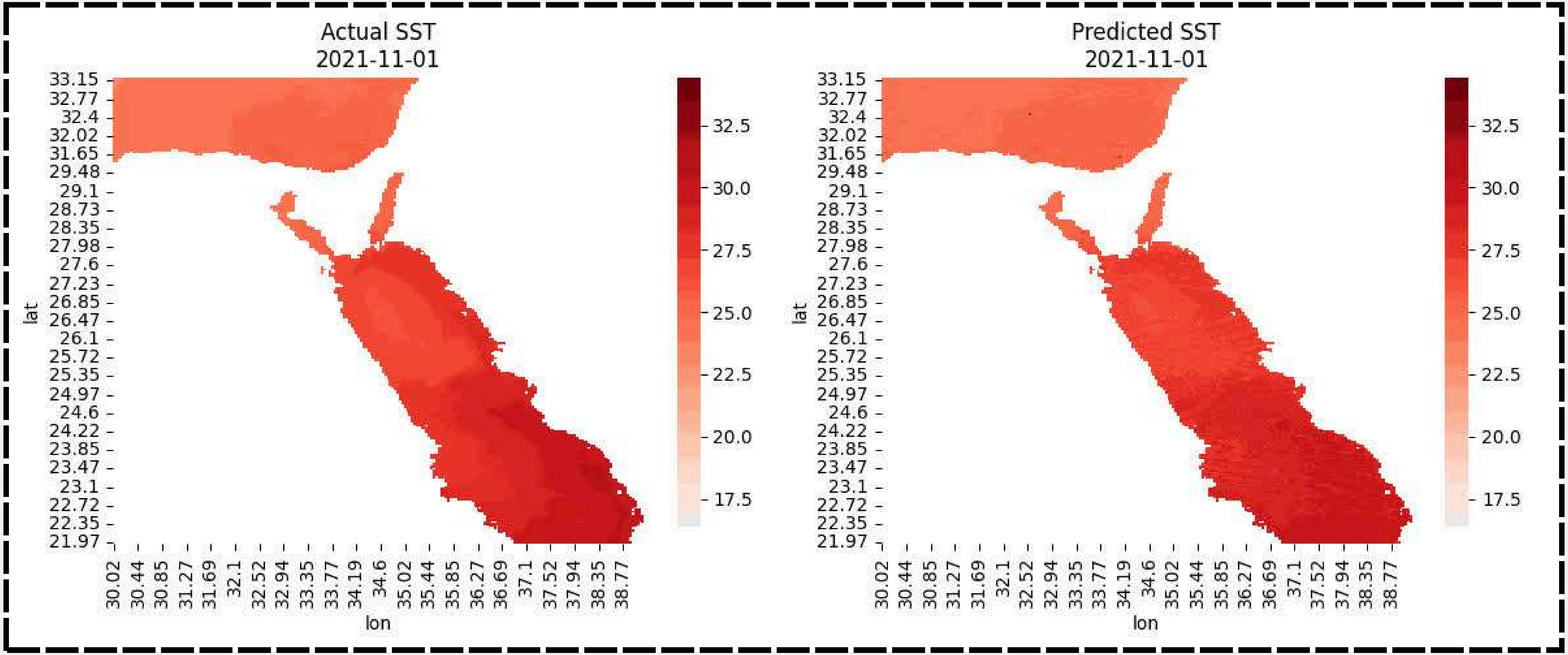
Model Development

VISUAL RESULTS – PREDICTED VS ACTUAL DISTRIBUTIONS





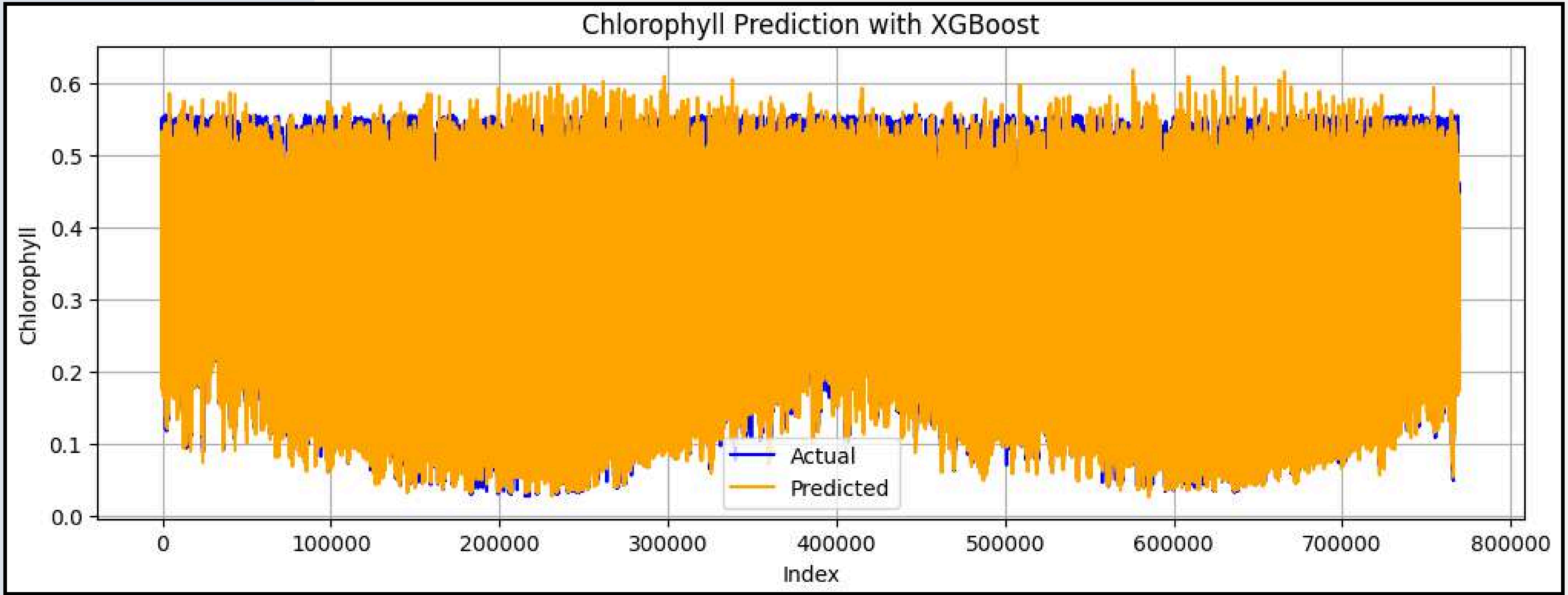
Model Development



Model Development

CHLOROPHYLL-A PREDICTION USING XGBOOST – FEATURE-RICH MODEL

To improve Chlorophyll-a prediction, we implemented an XGBoost regression model using a feature-rich, multi-dimensional dataset.



Model Development

CHLOROPHYLL-A PREDICTION USING XGBOOST – FEATURE-RICH MODEL

Key Features Used:

- Temporal Patterns:
  - chl_diff, chl_ma3, chl_std3
  - sst, sst_diff, sst_ma3
- Seasonal Encoding:
  - Month_sin, Month_cos
- Spatial Info:
  - Latitude, Longitude
- Target Transformation:
  - log1p(chlor_a) → stabilized wide range of values

Model Setup:

- Data split: 80% train / 20% test

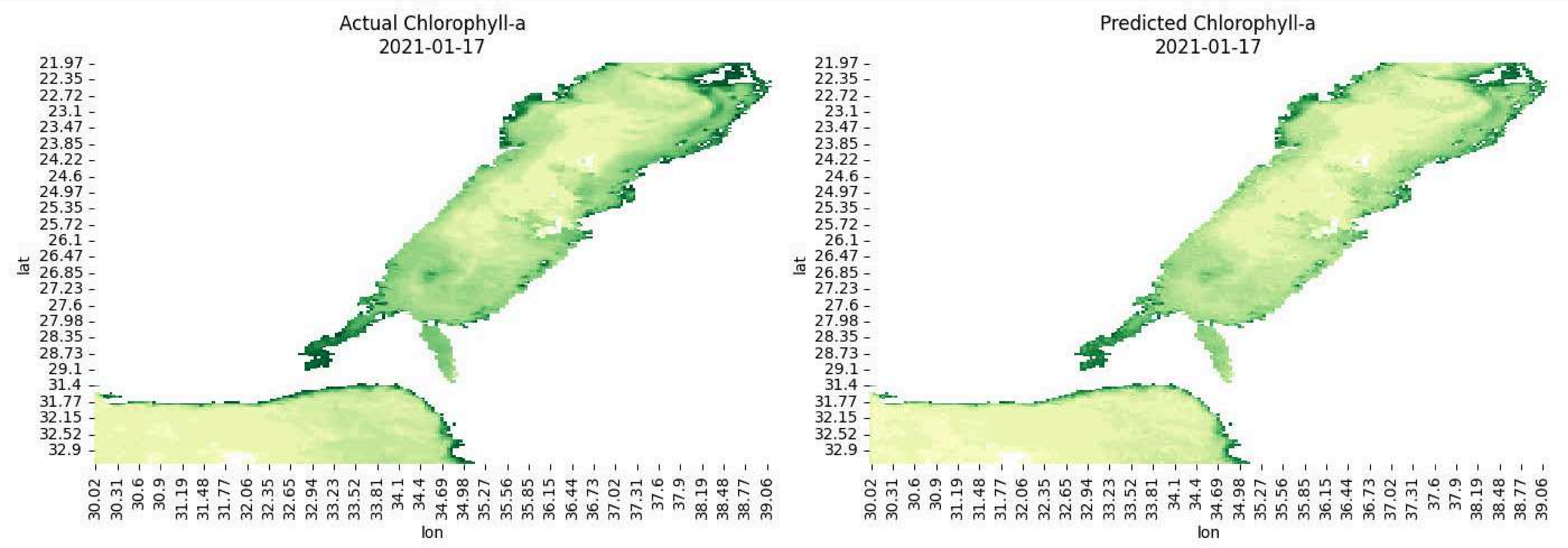
```
# ✅ Final Feature Set for Chlorophyll-a Prediction using XGBoost

features = [
    'chl_diff',     # Difference in Chlorophyll-a from the previous time step
    'chl_ma3',      # 3-step moving average of Chlorophyll-a (trend)
    'chl_std3',     # 3-step rolling standard deviation (variability)

    'sst',          # Sea Surface Temperature at current time
    'sst_ma3',      # 3-step moving average of SST (thermal trend)
    'sst_diff',     # SST change compared to previous value

    'lat',          # Latitude (spatial location)
    'lon',          # Longitude (spatial location)

    'Month_sin',    # Sine of month index (cyclical seasonality)
    'Month_cos'     # Cosine of month index (cyclical seasonality)
]
```

Performance

Metric	Value
R^2 Score	0.9744
MAE	0.0089
MSE	0.0003

Model Development

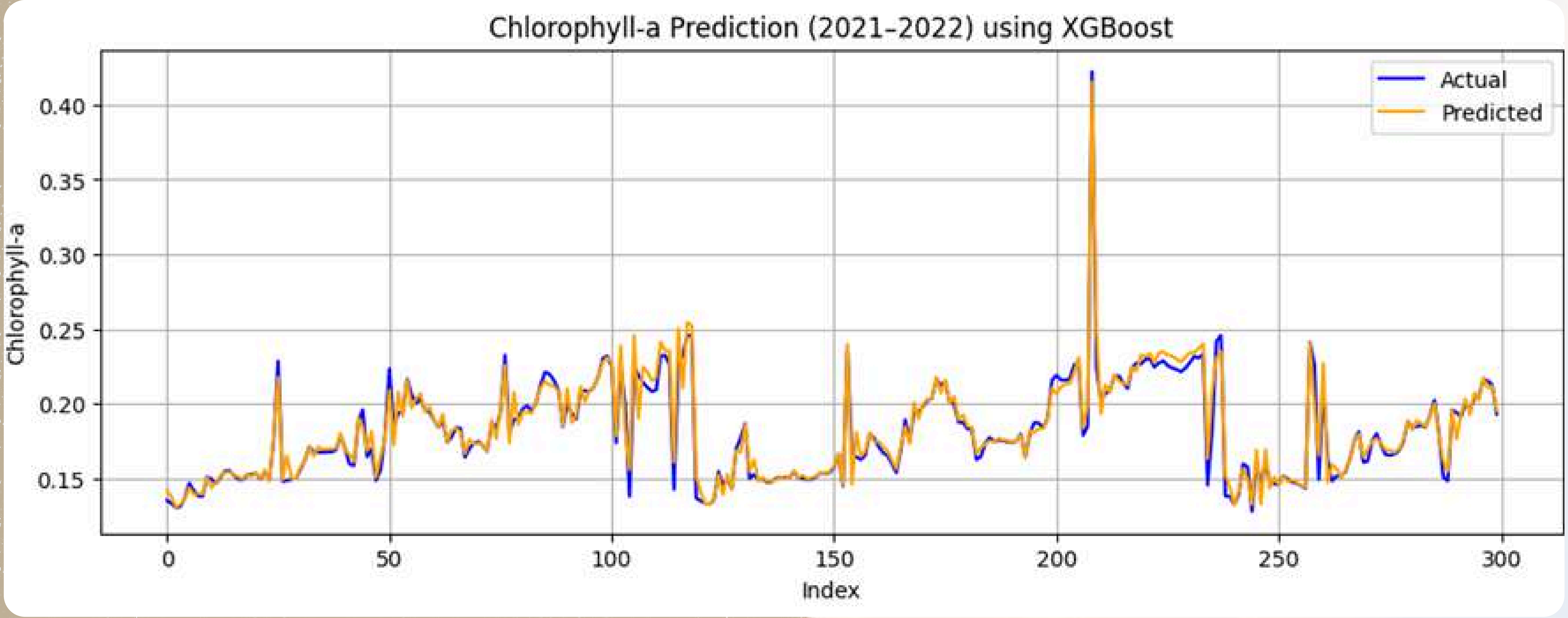
XGBOOST PREDICTIONS – SAMPLE & FINAL CONCLUSION



Temporal Validation - XGBoost Model on 2021 Data

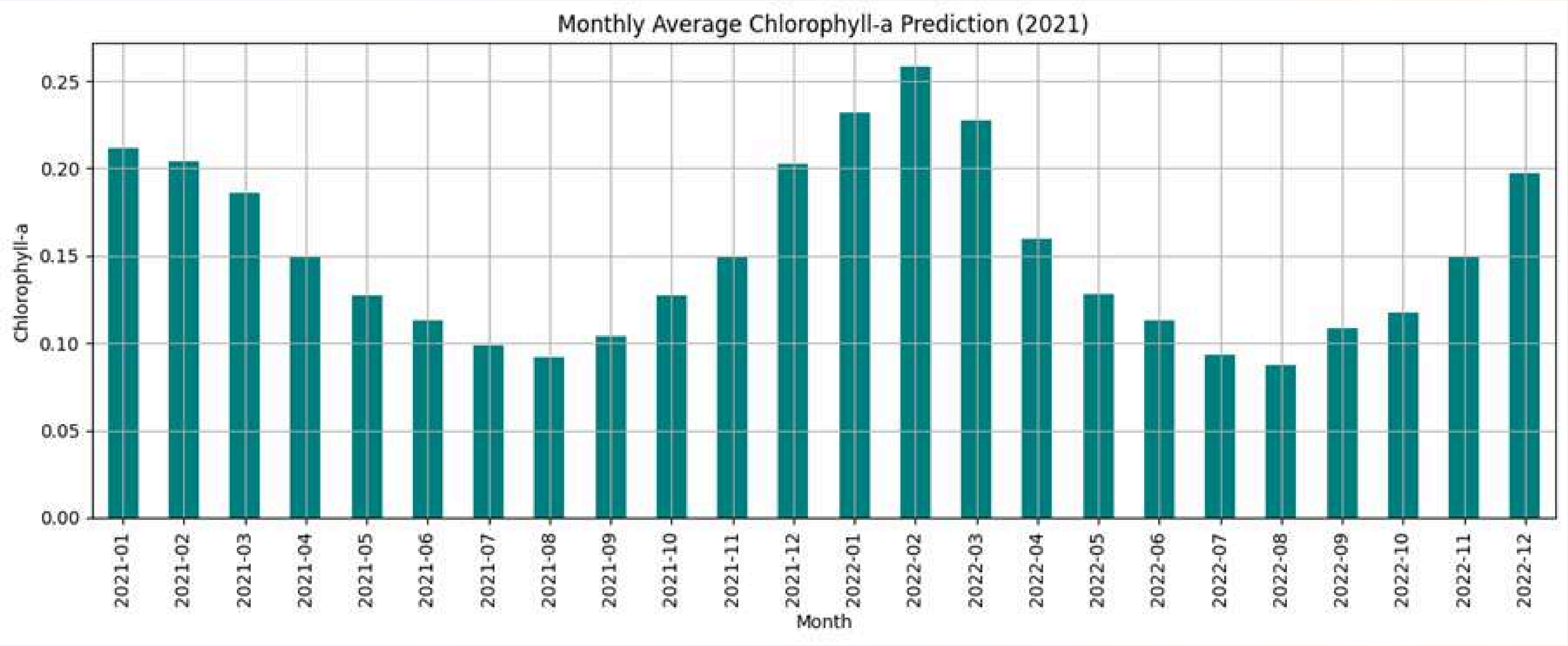
Validation Setup:
- Training Year: 2020
- Testing Year: 2021

Input: Engineered + scaled features (SST, Chlorophyll, Month, Lat/Lon)
Target: log1p(chlor_a) → inverse transformed after prediction



Temporal Validation - XGBoost Model on 2021 Data

Performance on 2021

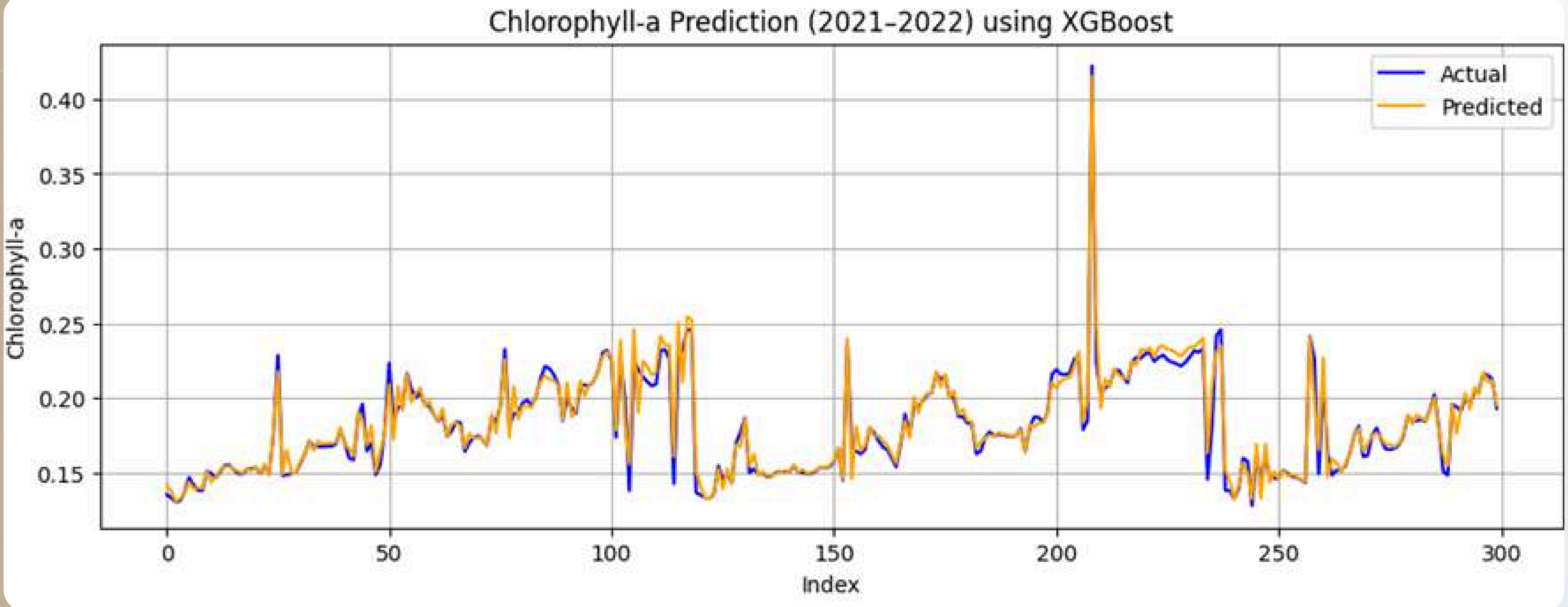


Metric	Value
R^2 Score	**0.9721**
MAE	0.0081
MSE	0.0002

Long-Term Validation: 2-Year Generalization of XGBoost Modelt Model on 2021 Data

Validation Setup:

- Training Year: 2020
- Validation Period: 2021 & 2022



Year vs 2-Year Validation - XGBoost Performance Comparison

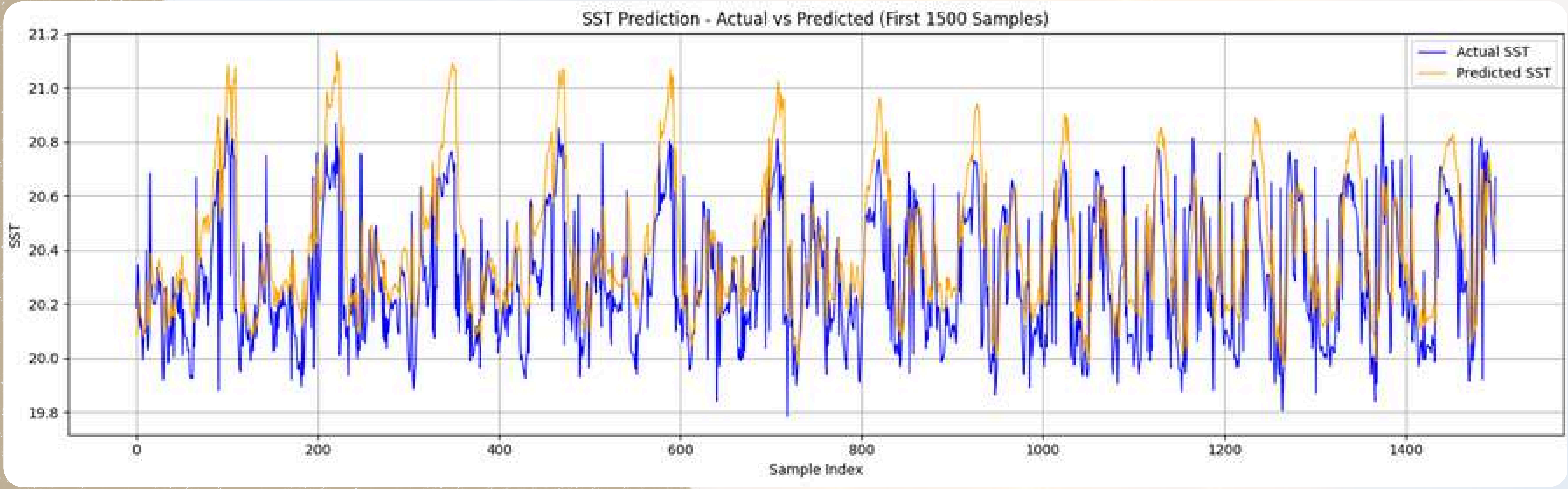
Key Insights:

1. Better accuracy over 2 years, not just short-term
2. Lower MAE and higher R^2 → More stable over time
3. Confirms the model learned seasonal + spatial dynamics, not just trends from 2020
4. Engineered features (rolling stats + cyclical encoding) improved long-term generalization

Metric	2021 Only	2021–2022
MSE	0.0002	0.0002
MAE	0.0081	**0.0064**
R^2 Score	0.9721	**0.9792**

LSTM Model - SST Validation on 2021-2022 Data

We validated the trained LSTM SST model using unseen satellite data from 2021–2022 to confirm long-term robustness.



Temporal Validation - XGBoost Model on 2021 Data

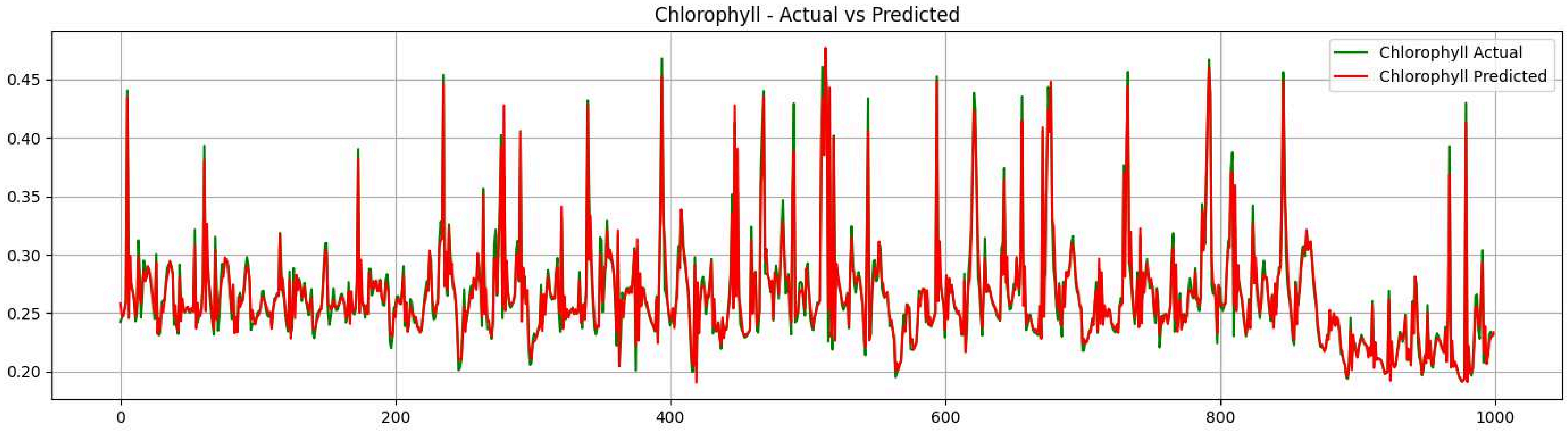
Performance Summary

Sample	Actual SST	Predicted SST	Residual (Actual – Predicted)
1	20.185	20.0868	0.0982
2	20.345	20.0932	0.2518
3	20.31	20.2207	0.0893
4	20.155	20.2591	-0.1041

Metric	Value
R^2 Score	**0.9747**
MAE	0.3267
MSE	0.326

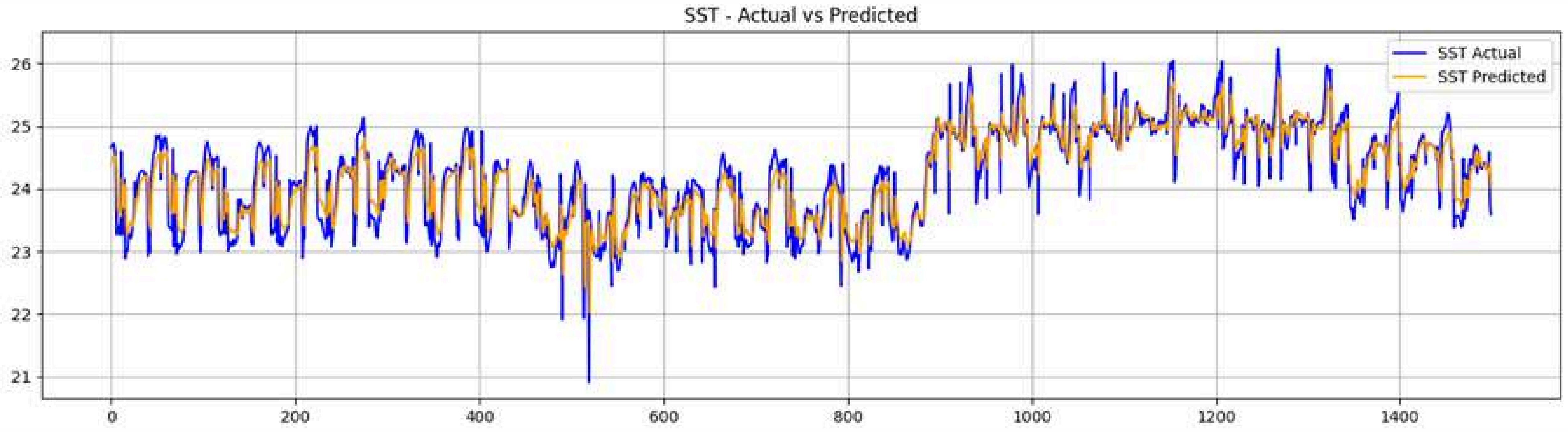
Real-Time Forecasting with 2025 Satellite Datault

To test the models in real-world conditions, we applied them to January–June 2025 satellite data, covering weekly SST and Chlorophyll-a observations.



Real-Time Forecasting with 2025 Satellite Datault

To test the models in real-world conditions, we applied them to January–June 2025 satellite data, covering weekly SST and Chlorophyll-a observations.



Performance on 2025 Data

	date	lat	lon	chlorophyll	sst
0	2025-05-24	32.94	34.27	0.049948	22.622613
1	2025-05-31	32.94	34.27	0.049818	22.458638
2	2025-06-07	32.94	34.27	0.049818	22.354344
3	2025-06-14	32.94	34.27	0.049818	22.263310
4	2025-06-21	32.94	34.27	0.049818	22.224109
5	2025-06-28	32.94	34.27	0.049818	22.194113
6	2025-07-05	32.94	34.27	0.049818	22.202601
7	2025-07-12	32.94	34.27	0.049818	22.216961
8	2025-07-19	32.94	34.27	0.049818	22.250343
9	2025-07-26	32.94	34.27	0.049818	22.286120
10	2025-08-02	32.94	34.27	0.049818	22.322352
11	2025-08-09	32.94	34.27	0.049818	22.332218
12	2025-08-16	32.94	34.27	0.049818	22.331132
13	2025-08-23	32.94	34.27	0.049818	22.320318
14	2025-08-30	32.94	34.27	0.049818	22.305744
15	2025-09-06	32.94	34.27	0.049818	22.290162
16	2025-09-13	32.94	34.27	0.049818	22.276625
17	2025-09-20	32.94	34.27	0.049818	22.266289
18	2025-09-27	32.94	34.27	0.049818	22.260169
19	2025-10-04	32.94	34.27	0.049818	22.257716
20	2025-10-11	32.94	34.27	0.049818	22.258296
21	2025-10-18	32.94	34.27	0.049818	22.260518
22	2025-10-25	32.94	34.27	0.049818	22.263083
23	2025-11-01	32.94	34.27	0.049818	22.264714
24	2025-11-08	32.94	34.27	0.049818	22.265028
25	2025-11-15	32.94	34.27	0.049818	22.263950

Model	R^2	MAE	MSE
Chlorophyll	0.9798	0.006	0.0001
SST	0.9762	0.248	0.1822

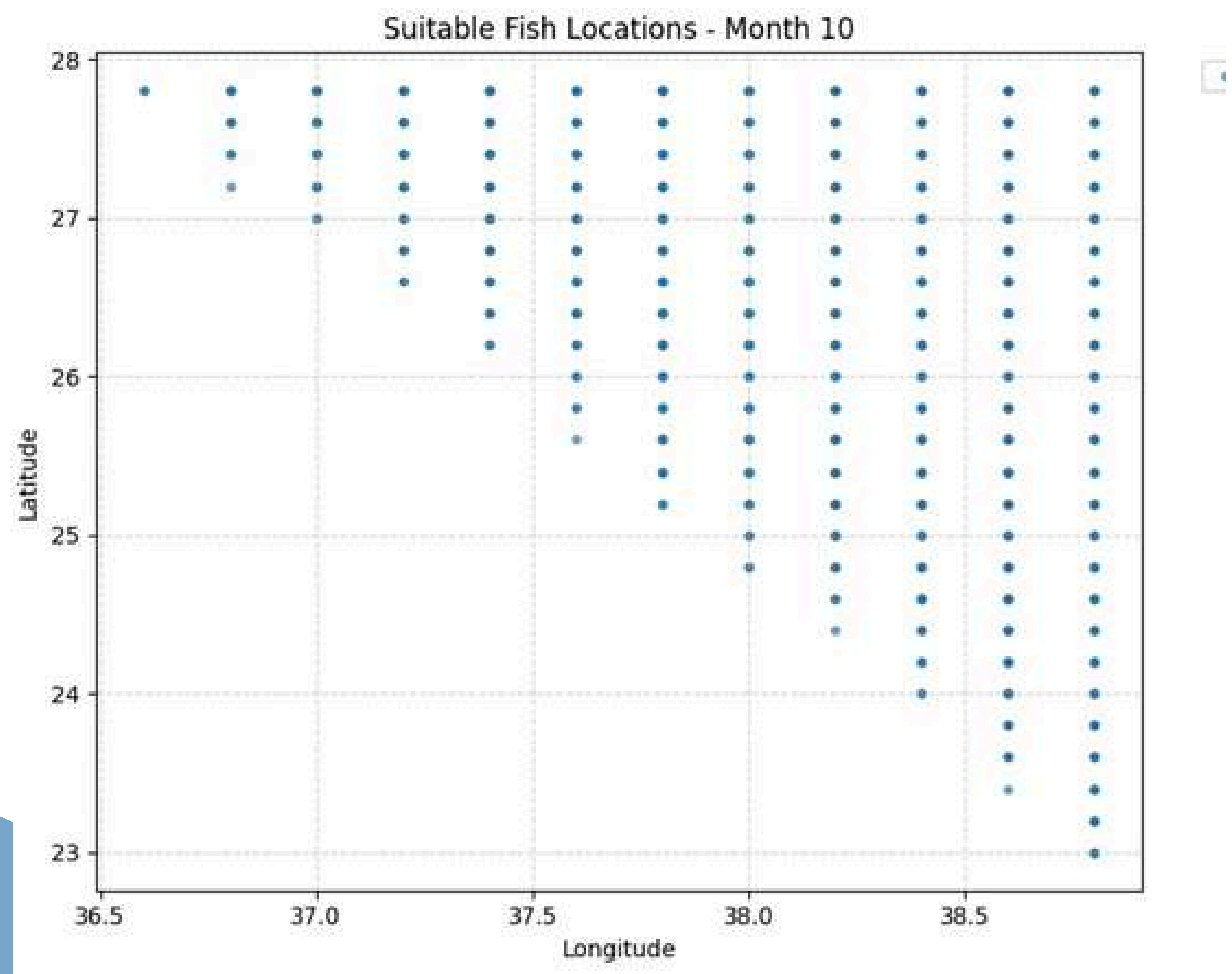
Fish Habitat Suitability Mapping – Jul to Dec 2025

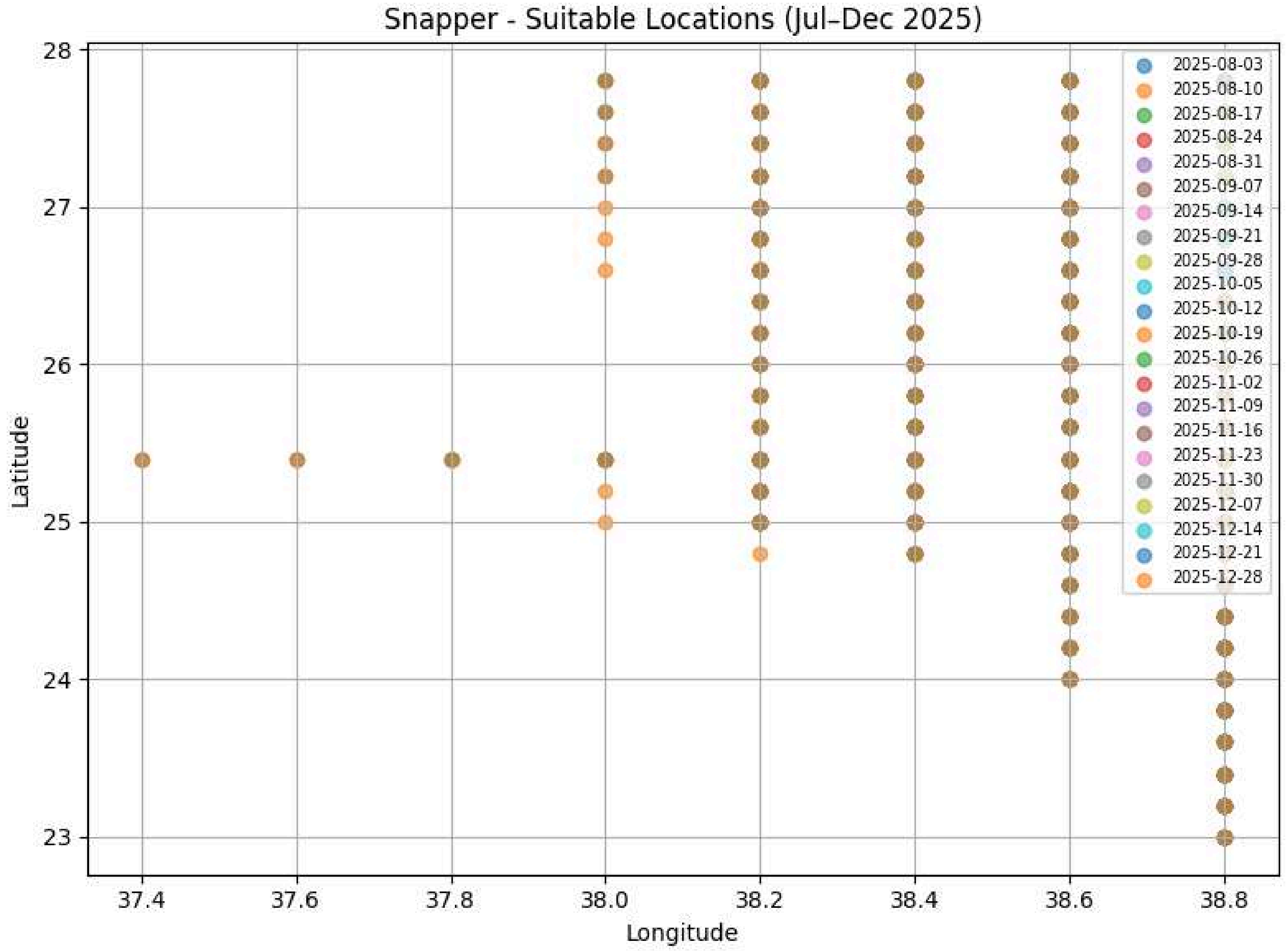
Using predicted SST and Chlorophyll-a values, we identified potential fish aggregation zones based on each species' ecological preferences.

Methodology Overview:

- Species Profiles:
  - **20 Red Sea fish species**
  - Each is defined by optimal SST and Chlorophyll-a range
- Weekly Filtering:
  - For each week (Jul–Dec 2025)
  - Extract locations where both SST & Chl-a match species' habitat conditions
- Visualization:
  - Generated weekly scatter maps
  - Each dot represents a potentially suitable zone for one species

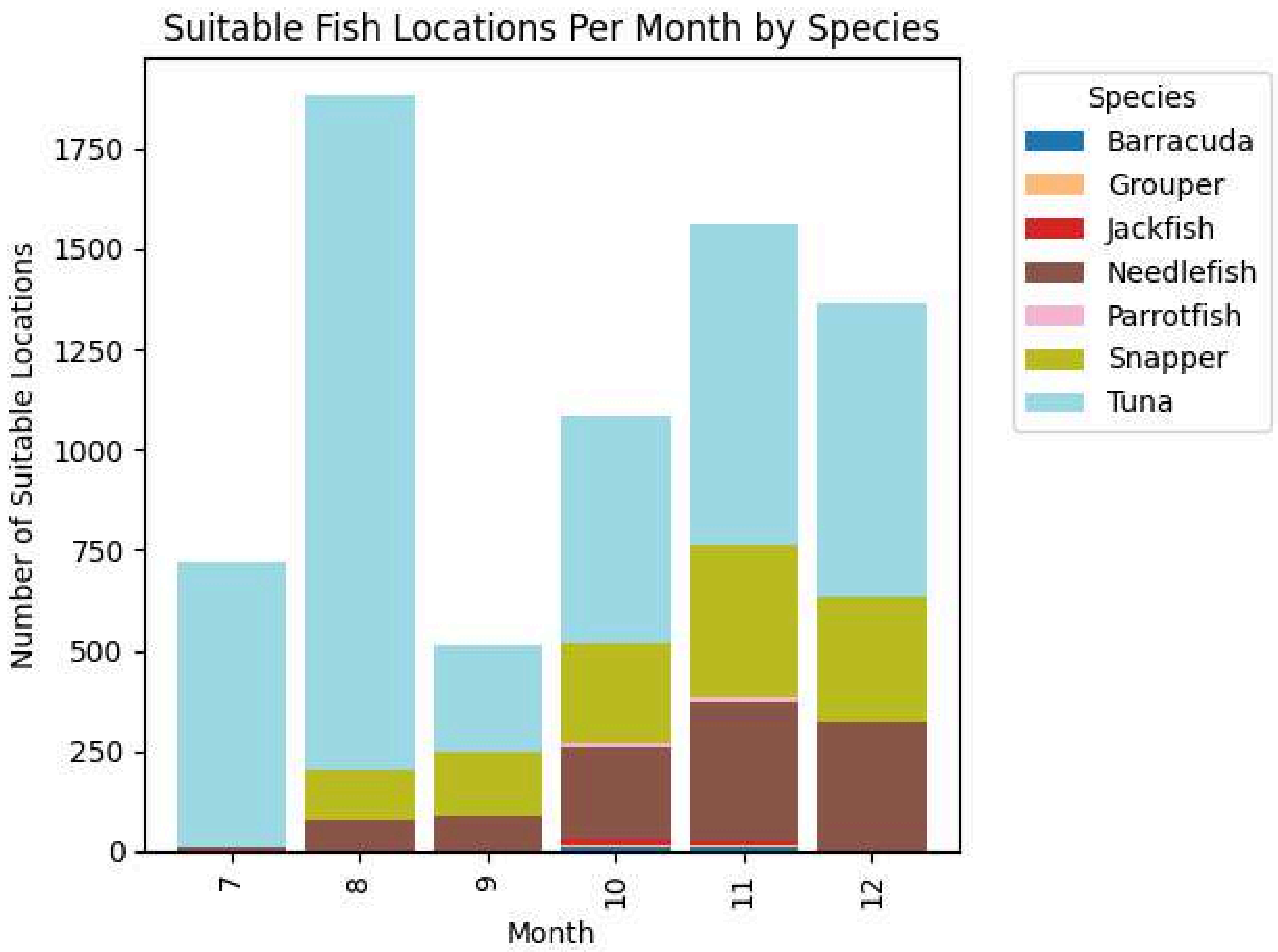
Output Highlight





Fish Distribution Analysis (Jul–Dec 2025)

Based on predicted SST and Chlorophyll-a, we analyzed the monthly suitability of habitats for 20 fish species across July–December 2025.



Final Conclusion – AI for Smarter & Sustainable Oceans

This project proved that satellite-based environmental data—specifically SST and Chlorophyll-a—can be used to predict biologically favorable fish aggregation zones in the Red Sea.

AI Models Used:

XGBoost → Accurate Chlorophyll-a prediction (R^2 = 0.9744)

LSTM → Reliable SST forecasting (R^2 = 0.9747)

What We Achieved:

- High-resolution weekly forecasts (Jul–Dec 2025)
- 20 species analyzed with ecological filtering
- Dynamic fish suitability maps to guide sustainable fishing
- A scalable, data-driven AI system ready for real-world deployment

Embedded Software Systems

From Sentinel Satellites to a Smart Prototype

- Sentinel satellites (by ESA) collect environmental data using multispectral and thermal sensors.
- These satellites monitor chlorophyll levels, SST, land cover, and more.
- The presented system simulates part of this behavior using low-cost embedded sensors.

Satellite

Sensors

Processed Dataset

AI model

Raspberry Pi as the Onboard Controller

- Raspberry Pi 4 was selected to simulate a satellite’s onboard computer.
- Offers camera support, GPIO interfaces, Python environment, and multitasking.
- Enables real-time data collection and coordination.

Raspberry Pi as the Onboard Controller

Feature	Raspberry Pi 4	Arduino Uno/Nano	STM32 Microcontrollers
Processing Power	High (Quad-core CPU)	Low (8-bit, 16 MHz)	Medium–High (32-bit)
OS Support	Full Linux OS	None	No OS / RTOS
Camera Support	Native (CSI)	Not Available	Advanced setup only
Python Compatibility	Full	Not supported	Partial (MicroPython)
Ideal Use Case	AI, imaging, multitasking	Simple control tasks	Real-time control

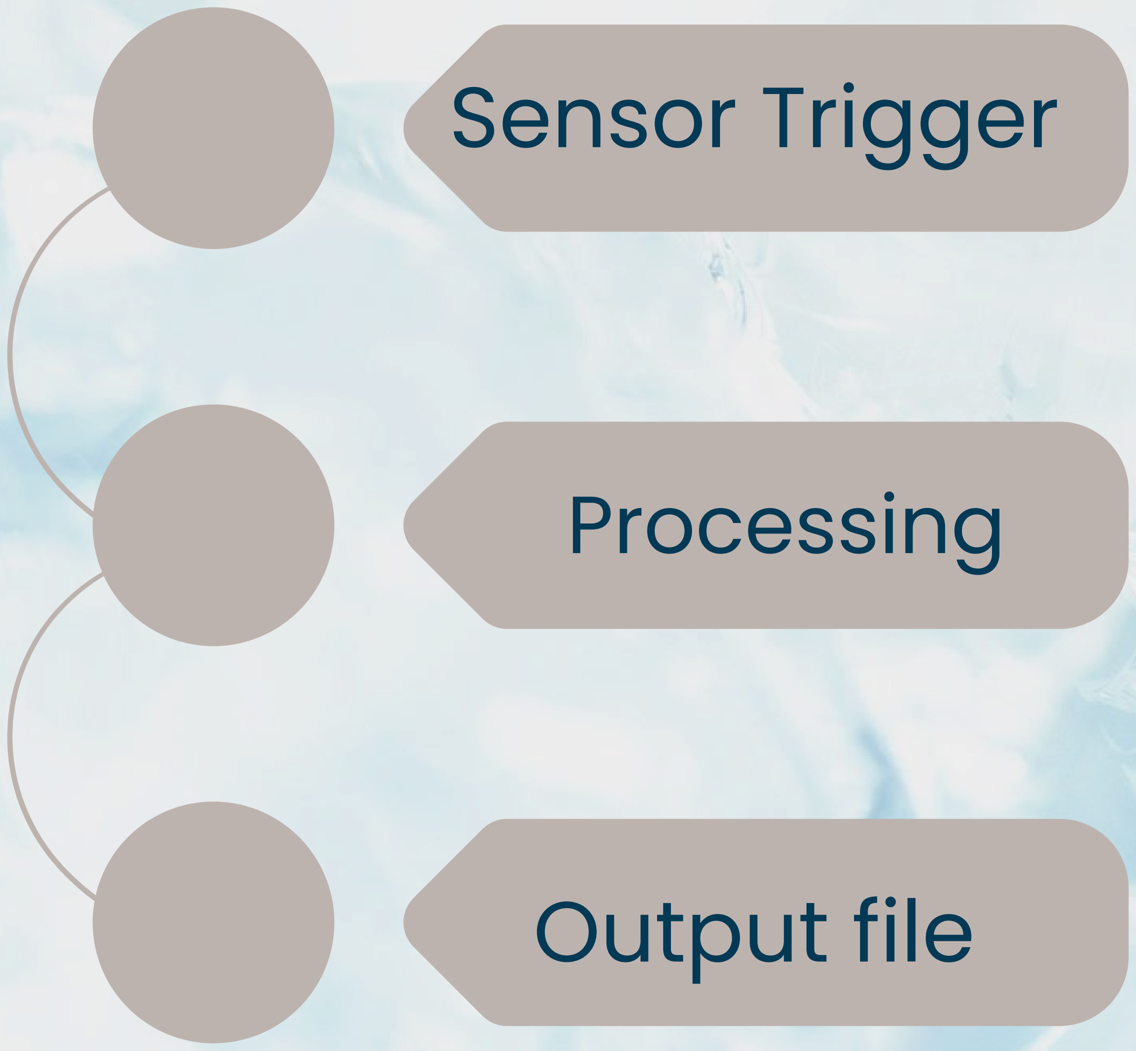
Simulated Sensor Suite

- Pi Camera → RGB images → NDVI → Chlorophyll estimation
- DHT22 → Air temperature → SST estimation
- GPS (NEO-6M) → Location tagging of sensor readings.

Sensor	Interface	GPIO Pins	Python Library	
DHT22	Digital GPIO	GPIO 4	adafruit-circuitpython-dht	
NEO-6M GPS	UART	GPIO 14 (TX), 15 (RX)	pynmea2, pyserial	
Pi Camera	CSI Ribbon	Camera Slot	libcamera, opencv-python	

Data Collection and Logging Pipeline

- Each sensor is controlled by a Python script.
- Captured data is processed, timestamped, and logged into a structured dataset.
- The output forms a synchronized CSV file with essential environmental indicators.



From Raw Data to Scientific Estimation

CHLOROPHYLL ESTIMATION USING RASPBERRY PI CAMERA AND NDVI APPROXIMATION

SEA SURFACE TEMPERATURE ESTIMATION USING DHT22 SENSOR

GPS LOGGING USING RASPBERRY PI AND NEO-6M MODULE

1.GPS LOGGING USING RASPBERRY PI AND NEO-6M MODULE

1.1 Output

GPS_data.csv

```
timestamp,latitude,longitude
2025-07-07 23:40:54,30.04954,31.607688
2025-07-07 23:40:56,30.049532,31.607685
2025-07-07 23:40:58,30.049536,31.607688
2025-07-07 23:41:00,30.049533,31.607692
2025-07-07 23:41:02,30.049544,31.607682
2025-07-07 23:41:04,30.049531,31.607684
2025-07-07 23:41:06,30.049543,31.607693
2025-07-07 23:41:08,30.049548,31.607688
2025-07-07 23:41:10,30.049546,31.607682
```

2.CHLOROPHYLL ESTIMATION USING RASPBERRY PI CAMERA AND NDVI APPROXIMATION

2.1 How calculated NDVI?

- NDVI (real):

$$\mathbf{NDVI} = \frac{(\mathbf{NIR} - \mathbf{Red})}{(\mathbf{NIR} + \mathbf{Red})}$$

- NDVI Approximation:

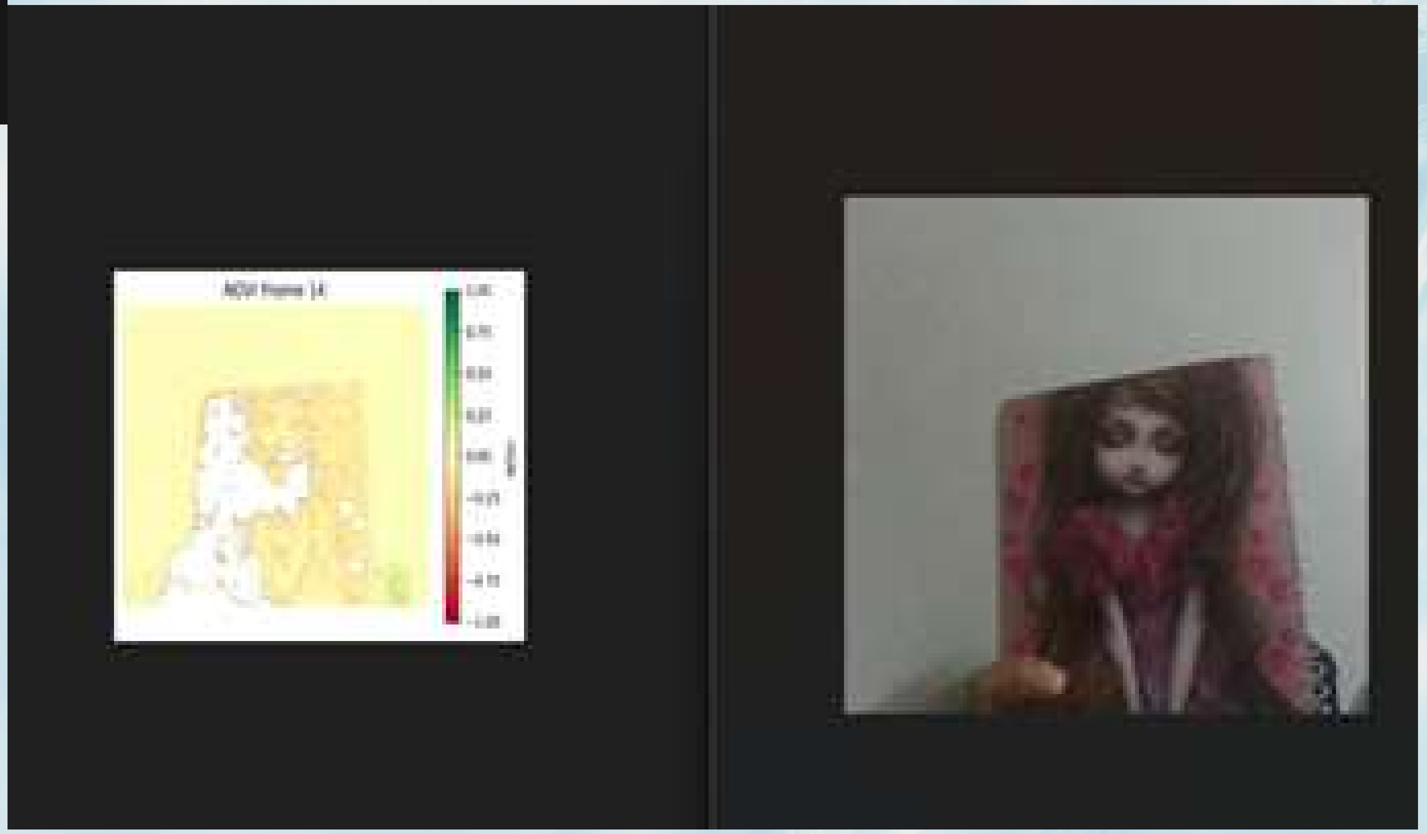
$$\mathbf{NDVI(approx)} = \frac{(\mathbf{Blue} - \mathbf{Red})}{(\mathbf{Blue} + \mathbf{Red} + \varepsilon)}$$

Where:

- Red and Blue represent pixel intensity values from the respective RGB channels.
- ε epsilon is a small positive constant added to prevent division by zero (typically ε= 10^-5)

2.CHLOROPHYLL ESTIMATION USING RASPBERRY PI CAMERA AND NDVI APPROXIMATION

```
red = frame[:, :, 2].astype(float)
nir = frame[:, :, 0].astype(float)  # Using blue as NIR approximation
ndvi = (nir - red) / (nir + red + 1e-5)

ndvi_filtered = np.where(ndvi < -0.2, np.nan, ndvi)
ndvi_valid = ndvi_filtered[~np.isnan(ndvi_filtered)]
avg_ndvi = np.mean(ndvi_valid)
```



2.CHLOROPHYLL ESTIMATION USING RASPBERRY CAMERA AND NDVI APPROXIMATION

2.2 From NDVI approximation estimated chlorophyll-a.

This equation was integrated into the software and implemented in Python as:

```
chlor_a = 3.5106 * (avg_ndvi ** 2) + 8.3298 * avg_ndvi + 0.601
```

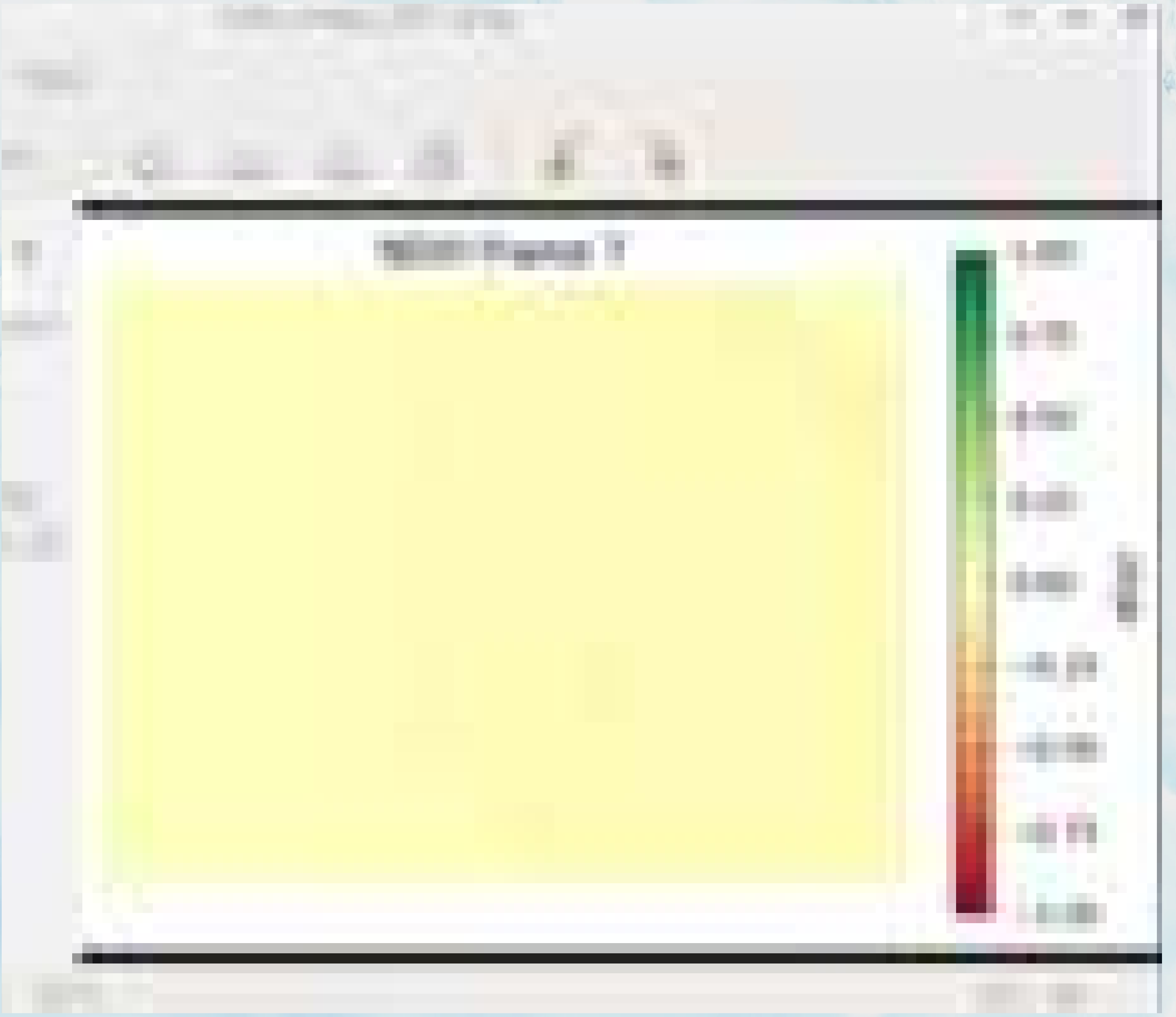
The applied equation is as follows:

$$Chlorophyll\text{-}a = a.(NDVI)^2 + b.NDVI + c$$

Where coefficients a, b, and c are derived from scientific reference" *Bung Binh Thien Lake in southern Vietnam, which established a statistical relationship between NDVI and chlorophyll-a concentrations derived from Landsat 8 imagery (Nguyen et al., 2020)*".

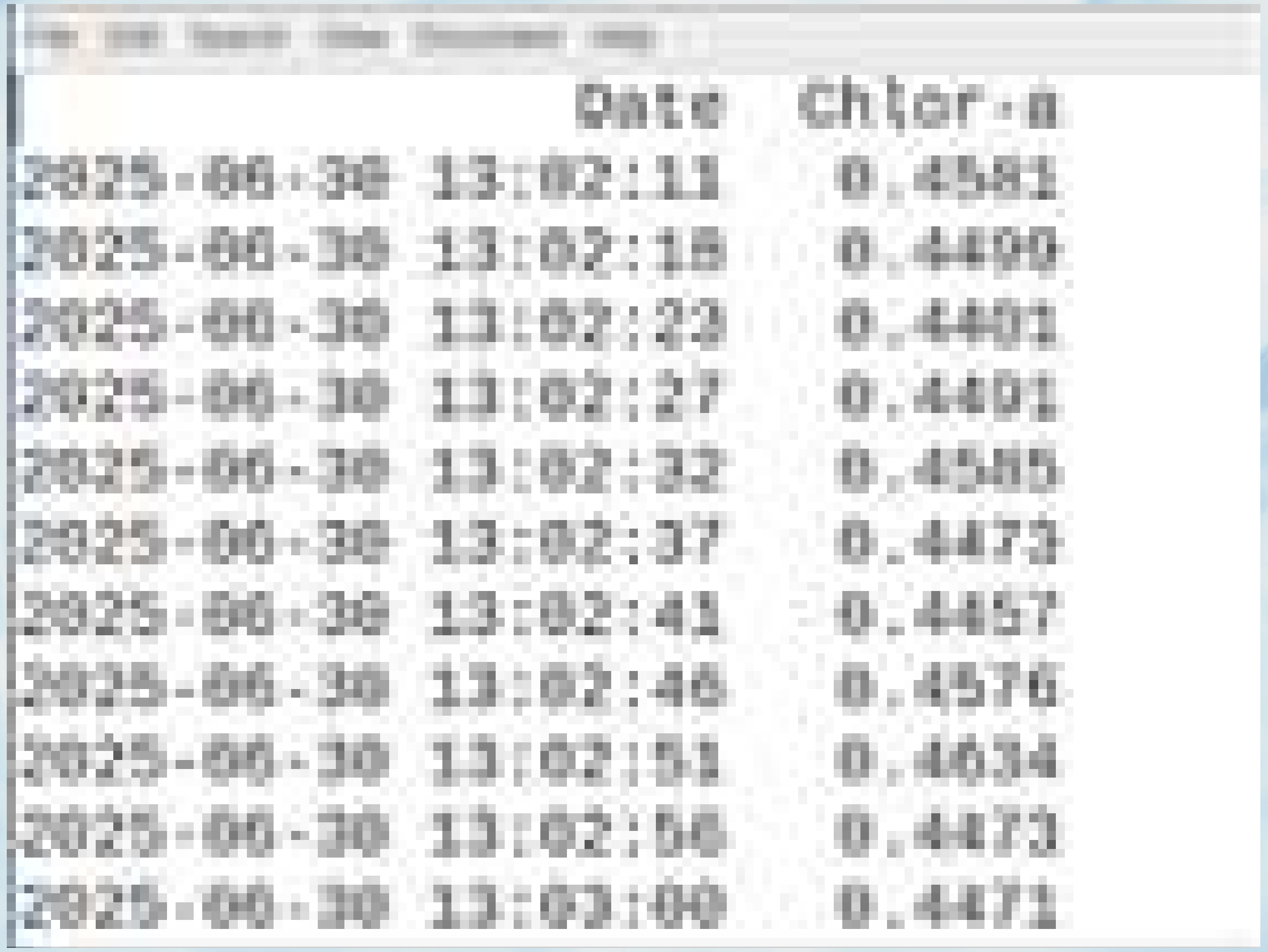
2.CHLOROPHYLL ESTIMATION USING RASPBERRY PI CAMERA AND NDVI APPROXIMATION

2.3 Output

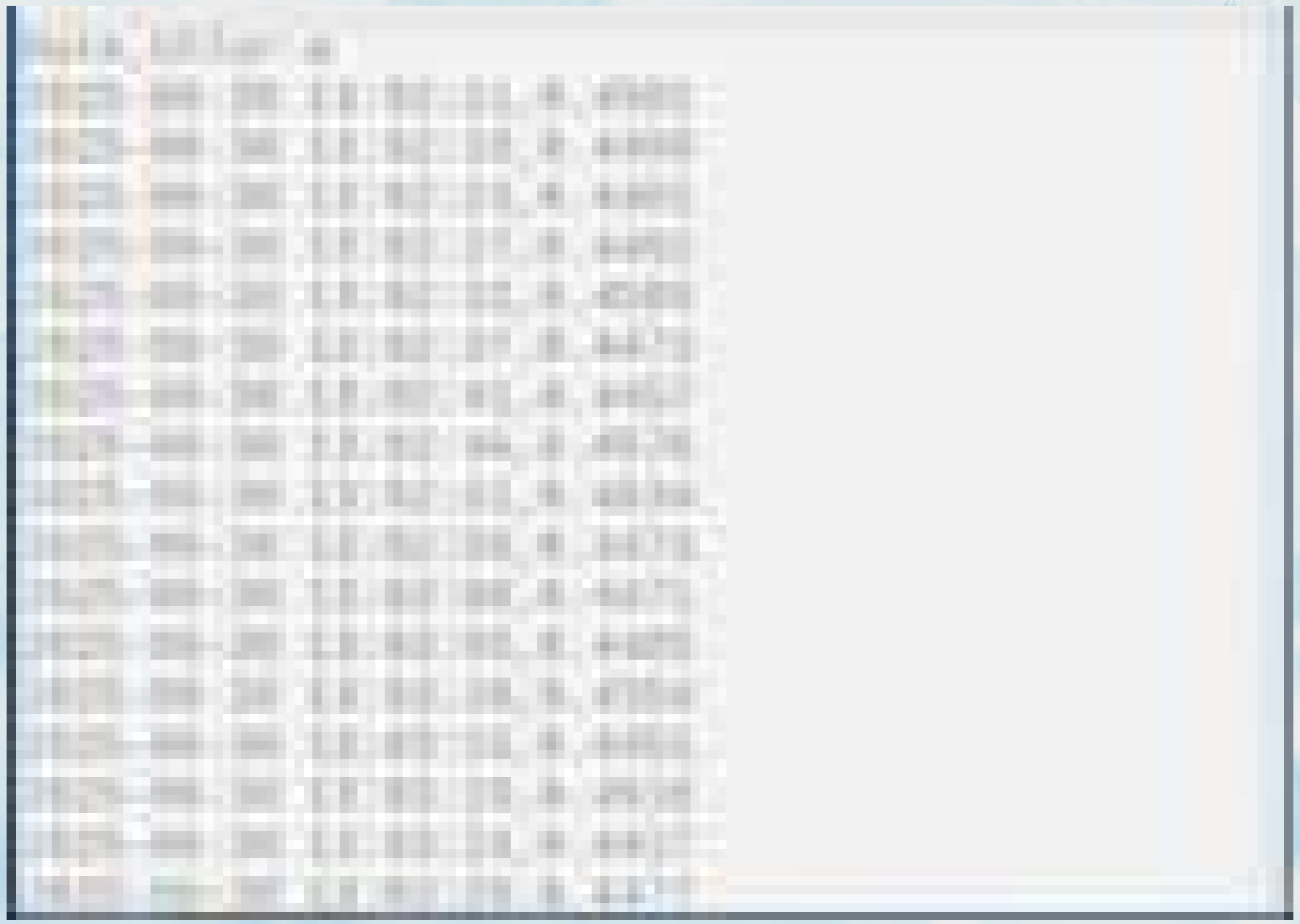


2.CHLOROPHYLL ESTIMATION USING RASPBERRY PI CAMERA AND NDVI APPROXIMATION

2.3 Output



Date	Chlor-a
2025-06-30 13:02:11	0.4581
2025-06-30 13:02:18	0.4490
2025-06-30 13:02:23	0.4401
2025-06-30 13:02:27	0.4491
2025-06-30 13:02:32	0.4585
2025-06-30 13:02:37	0.4473
2025-06-30 13:02:41	0.4457
2025-06-30 13:02:46	0.4576
2025-06-30 13:02:51	0.4634
2025-06-30 13:02:56	0.4473
2025-06-30 13:03:00	0.4471



3. SEA SURFACE TEMPERATURE ESTIMATION USING DHT22 SENSOR

3.1 How is DHT22 used to estimate SST?

SST Estimation (El-Geziry et al. 2023):

if $T\_air < 24°C$:

$$SST = 0.3832 \times T\_air + 12.154$$

else:

$$SST = 0.6567 \times T\_air + 5.4271$$

```python
def estimate_sst(temp_air):
    if temp_air < 24:  # Approximate winter condition
        return 0.3832 * temp_air + 12.154
    else:  # Approximate summer condition
        return 0.6567 * temp_air + 5.4271
```

3. SEA SURFACE TEMPERATURE ESTIMATION USING DHT22 SENSOR

3. 2 Output

SST_log_2025-07-06_16-34-21 > sst_readings.csv

```
Timestamp,Air_Temperature_C,Estimated_SST_C
2025-07-06 16:34:21,22.47,20.76
2025-07-06 16:34:23,22.8,20.89
2025-07-06 16:34:25,23.13,21.02
2025-07-06 16:34:27,22.74,20.87
2025-07-06 16:34:29,22.25,20.68
2025-07-06 16:34:31,22.85,20.91
2025-07-06 16:34:33,22.21,20.66
2025-07-06 16:34:35,22.92,20.94
2025-07-06 16:34:37,22.79,20.89
```

Structured Environmental Dataset

- CSV entries include:
  – Timestamp
  – Latitude / Longitude
  – SST
  – Chlorophyll-a value

sensors_data.csv

sensors_data_2025-07-06_16-28-46 > sensors_data.csv

```
1  Date,Latitude,Longitude,SST,Chlor-a
2  2025-07-06 16:28:47,30.049547,31.607697,21.14,-0.1164
3  2025-07-06 16:28:49,30.049542,31.607683,21.28,1.0965
4  2025-07-06 16:28:52,30.049548,31.607689,21.14,0.8163
5  2025-07-06 16:28:54,30.049535,31.607696,21.35,1.1043
6  2025-07-06 16:28:56,30.049536,31.607687,21.16,1.1488
7  2025-07-06 16:28:58,30.049533,31.607688,21.19,1.2138
```

- This dataset was used to feed the AI fish detection model.

Hardware Constraints and Solutions

Limitation	Workaround	Functional Result
No NIR sensor	Used Blue channel as NIR proxy	NDVI approximation calculated
No dedicated SST sensor	Used DHT22 + empirical SST model	SST values estimated from air temperature
GPS gives current ground location	Used fixed/simulated coordinates	Data still geo-tagged realistically

Enabling the AI-Interface Connection

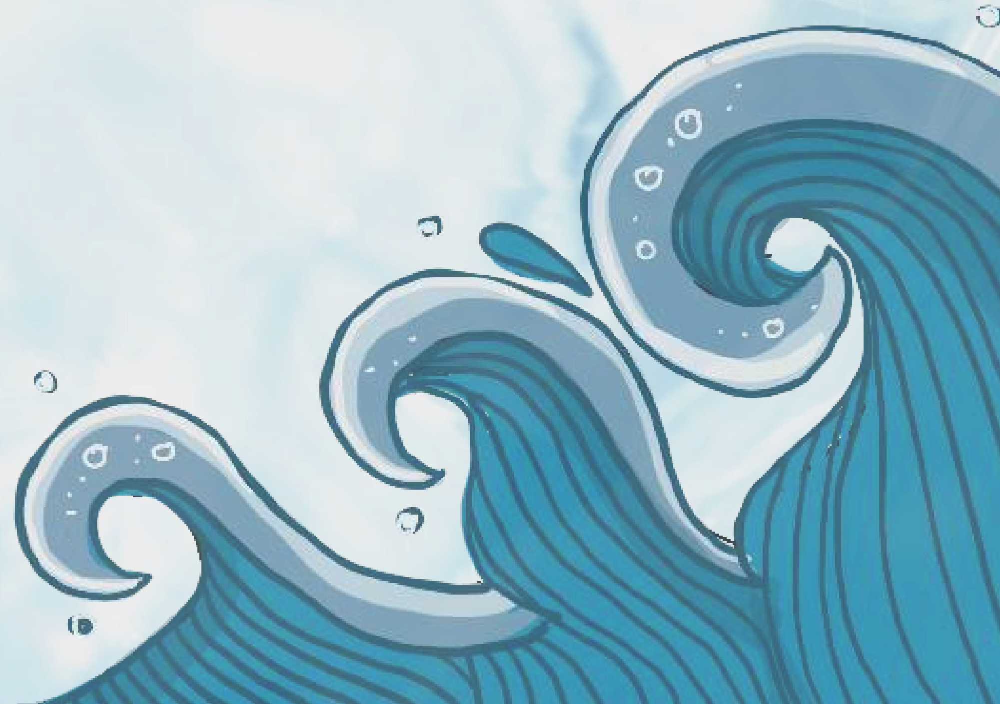
- The structured dataset enables the AI and interface systems to function effectively.
- Upcoming segment will explain how the data is integrated into real-time fish zone prediction.

SENSORS DATA

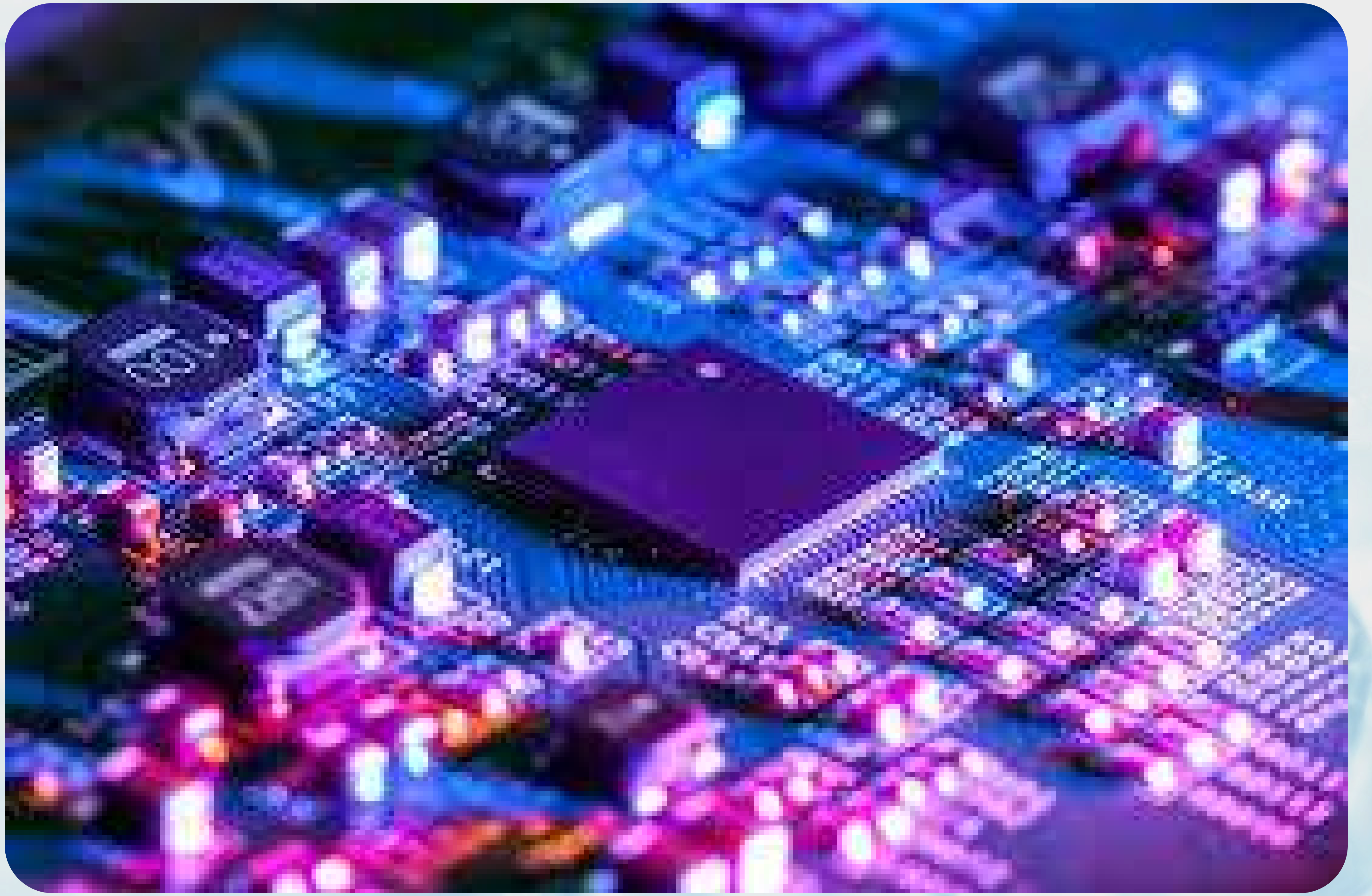
CSV

AI PREDICATION

MAP OUTPUT



Power and PCB Design



System Components Overview

Sensors & Camera

Camera v2:
8MP (Sony IMX219), connected via CSI
→ Image capture for monitoring & processing
→ Low power (250mA)

DHT22:
Measures temperature & humidity
→ Digital output
→ High accuracy, low current (~1.5mA)

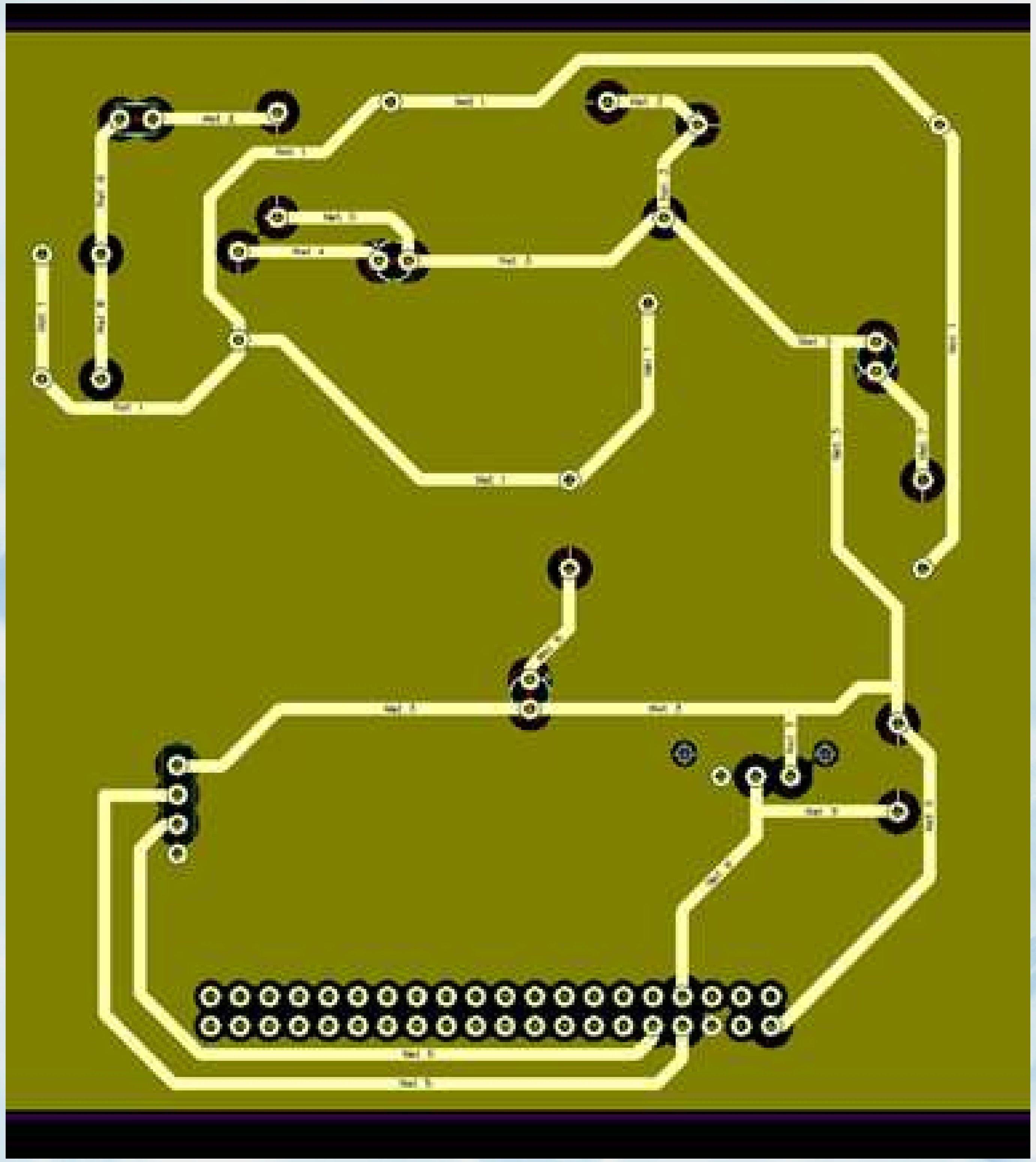
GPS Module (NEO-6M):
Provides real-time location (lat/lon/alt)
→ UART interface, 5V logic
→ Clean power needed for satellite lock

Raspberry Pi 4

Acts as the main controller
Handles image capture, sensor readings & data logging
Quad-core 1.5GHz CPU with up to 8GB RAM
Interfaces: GPIO, USB, CSI, UART, HDMI, Wi-Fi
Powered via 5V/2.6A USB-C

Power & Protection

Source: Redmi 10000mAh Power Bank
→ 5V @ 2.6A output
→ Portable & safe for testing
Design:
Unified 5V rail to all components
→ No voltage conversion needed
→ Capacitors for noise filtering
→ Fuse added for basic protection



PCB Design & Implementation

PCB Highlights:

- Single-layer 94×94 mm board
- Designed with DipTrace
- Wide 5V & GND traces
- Manual jumpers used

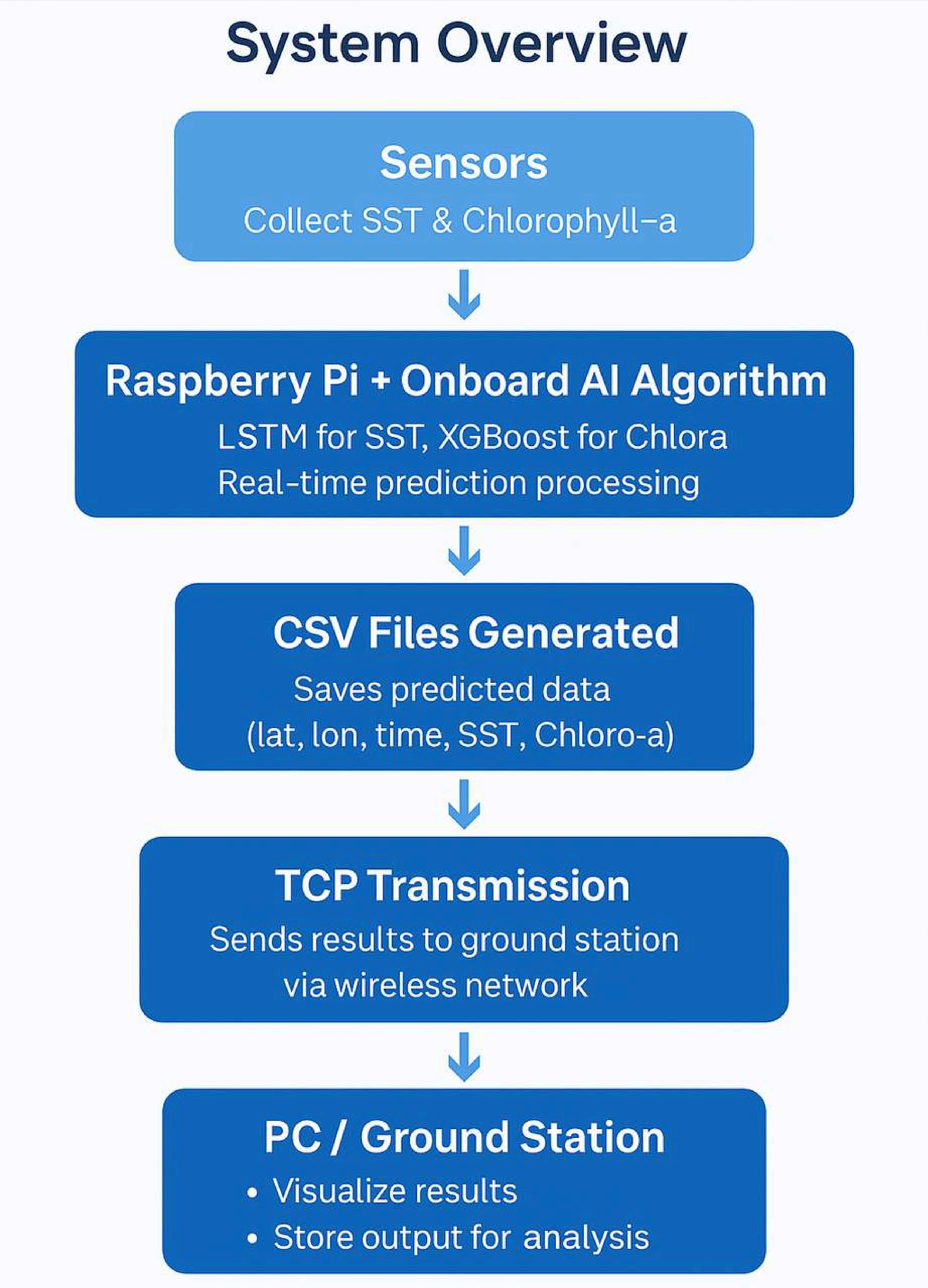
Protection Features:

- Fuse for overcurrent
- Decoupling capacitors near modules

Power Setup:

- All components powered by a 5V power bank
- No voltage regulators needed

ONBOARD AI INTERFACE


System Overview
Sensors
Collect SST & Chlorophyll–a
Raspberry Pi + Onboard AI Algorithm
LSTM for SST, XGBoost for Chlora
Real-time prediction processing
CSV Files Generated
Saves predicted data
(lat, lon, time, SST, Chloro-a)
TCP Transmission
Sends results to ground station
via wireless network
PC / Ground Station
• Visualize results
• Store output for analysis

- Sensors collect environmental and location data.
- AI algorithm runs onboard on Raspberry Pi
- AI classifies fish and calculates SST & chlorophyll
- Generates CSV files (lat, lon, SST, chl, fish type)
- CSV files are transmitted to the PC via TCP transmission

Overview of the onboard data processing flow from sensors to CSV generation and transmission

	Date	Latitude	Longitude	SST	Chlor-a	Fish_Type	Est_Quantity
1	Date	Latitude	Longitude	SST	Chlor-a	Fish_Type	Est_Quantity
2	########	30.04954	31.6077	26.86	2.6862	None	0
3	########	30.04953	31.60769	26.41	2.4486	None	0
4	########	30.04955	31.60769	26.38	1.6462	None	0
5	########	30.04954	31.60769	26.85	1.9272	None	0
6	########	30.04954	31.60769	26.38	2.9256	None	0
7	########	30.04954	31.60768	26.66	3.7574	None	0
8	########	30.04955	31.60769	26.44	4.8176	None	0
9	########	30.04953	31.60769	26.65	3.0594	None	0
10	########	30.04955	31.60769	26.27	3.0755	None	0
11	########	30.04955	31.60769	26.55	2.5489	None	0
12	########	30.04954	31.6077	26.58	0.4401	None	0
13	########	30.04954	31.60768	26.55	0.4579	None	0
14	########	30.04953	31.60768	26.84	0.3287	Marlin	25
15	########	30.04953	31.60768	26.7	0.3604	Marlin	22
16	########	30.04954	31.60769	26.72	0.568	None	0
17	########	30.04954	31.60769	26.33	0.3764	Barracuda	28
18	########	30.04955	31.60768	26.66	0.5119	None	0
19	########	30.04954	31.60768	26.43	0.4984	Barracuda	0
20	########	30.04953	31.60769	26.49	1.7285	None	0
21	########	30.04954	31.6077	26.76	0.9453	None	0

- ArcGIS for offline spatial mapping
- Python-based GUI for local interactive use
- Web-based visualization with Firebase integration

ARCGIS VISUALIZATION

ArcGIS Visualization

- CSV files generated onboard by Raspberry Pi were imported into ArcGIS
- Visualized fish locations using points and heatmaps
- Planned to use ArcGIS server to upload data continuously

"Point-based display of fish detection results imported from onboard-generated CSV."

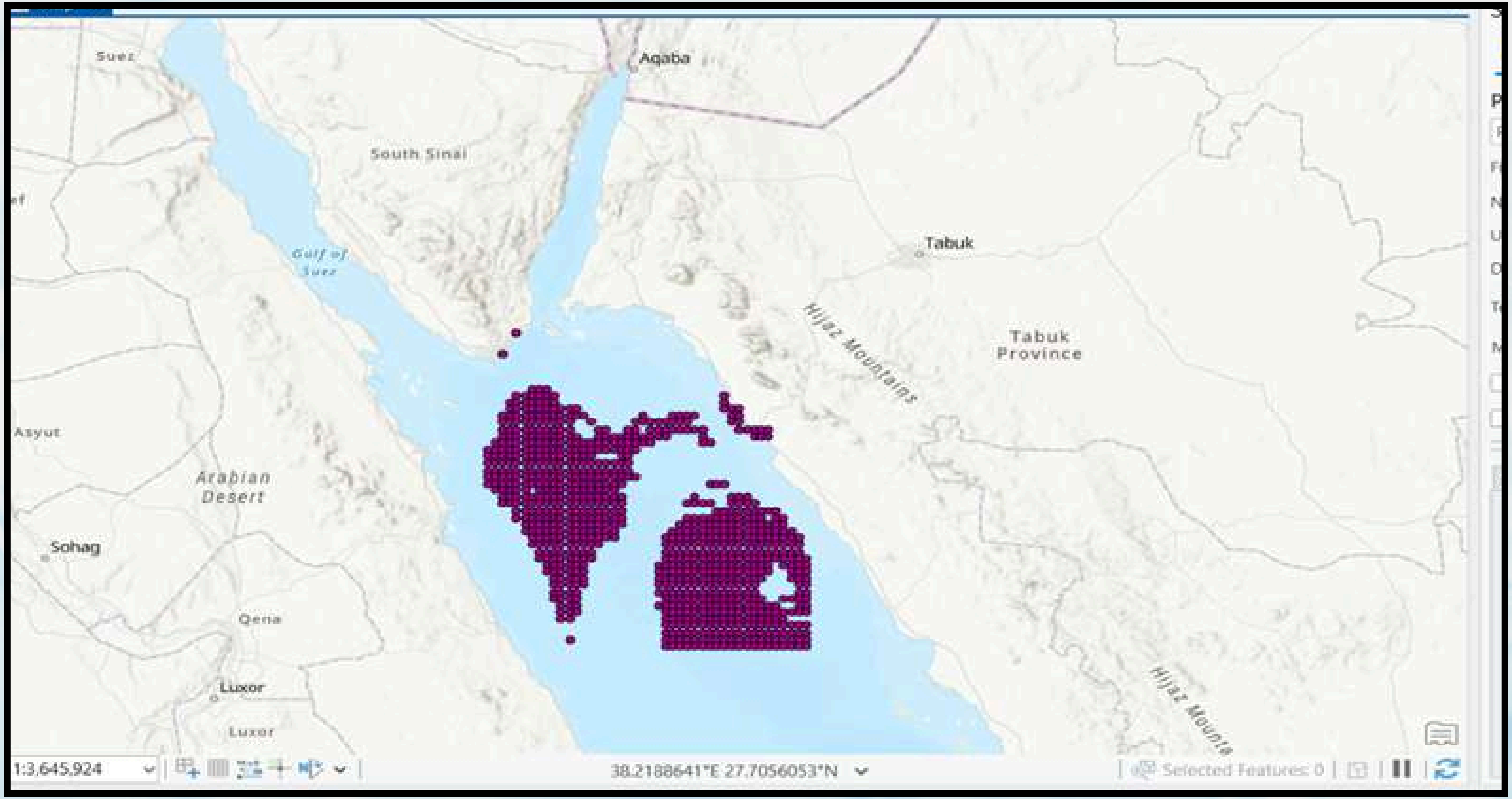
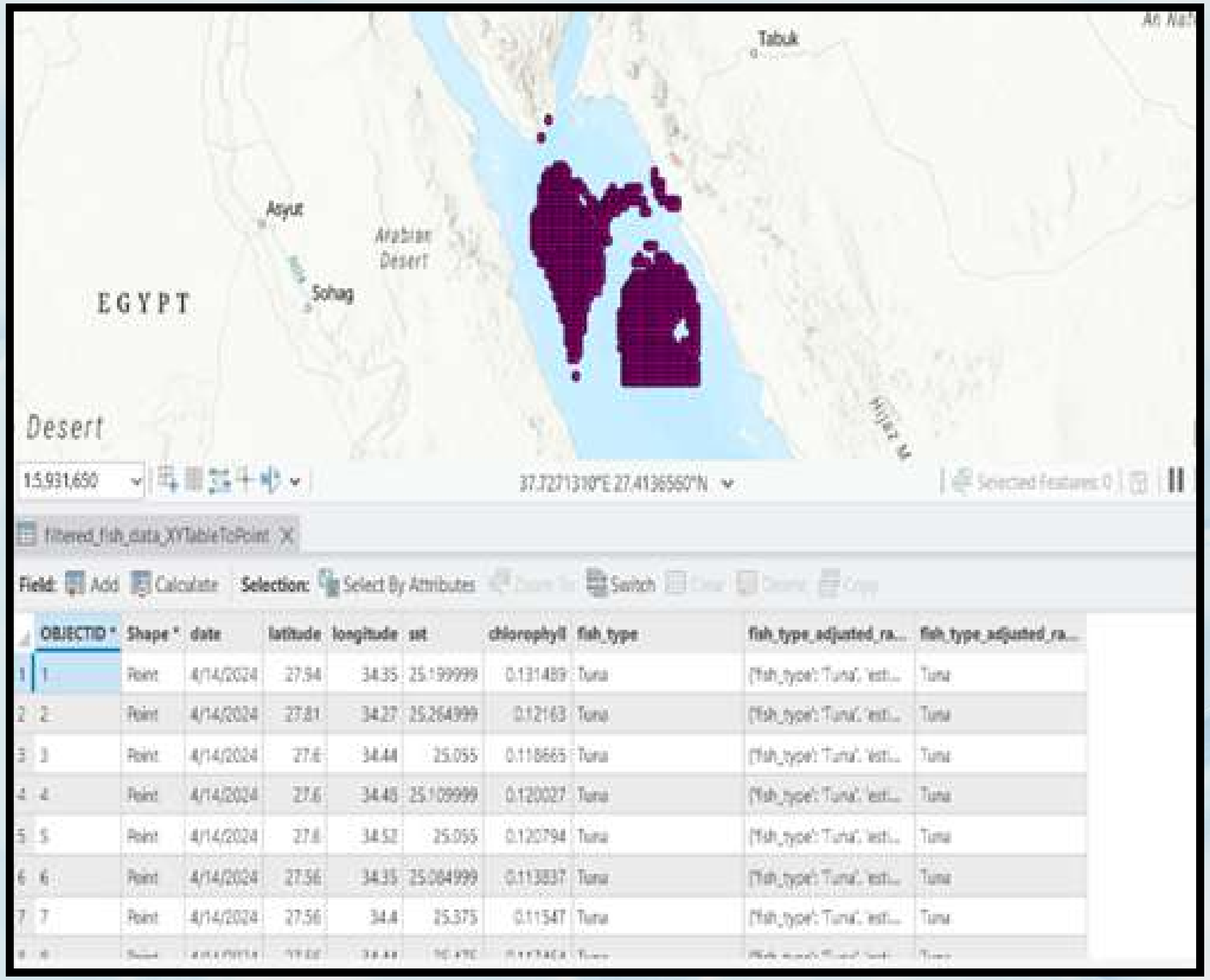
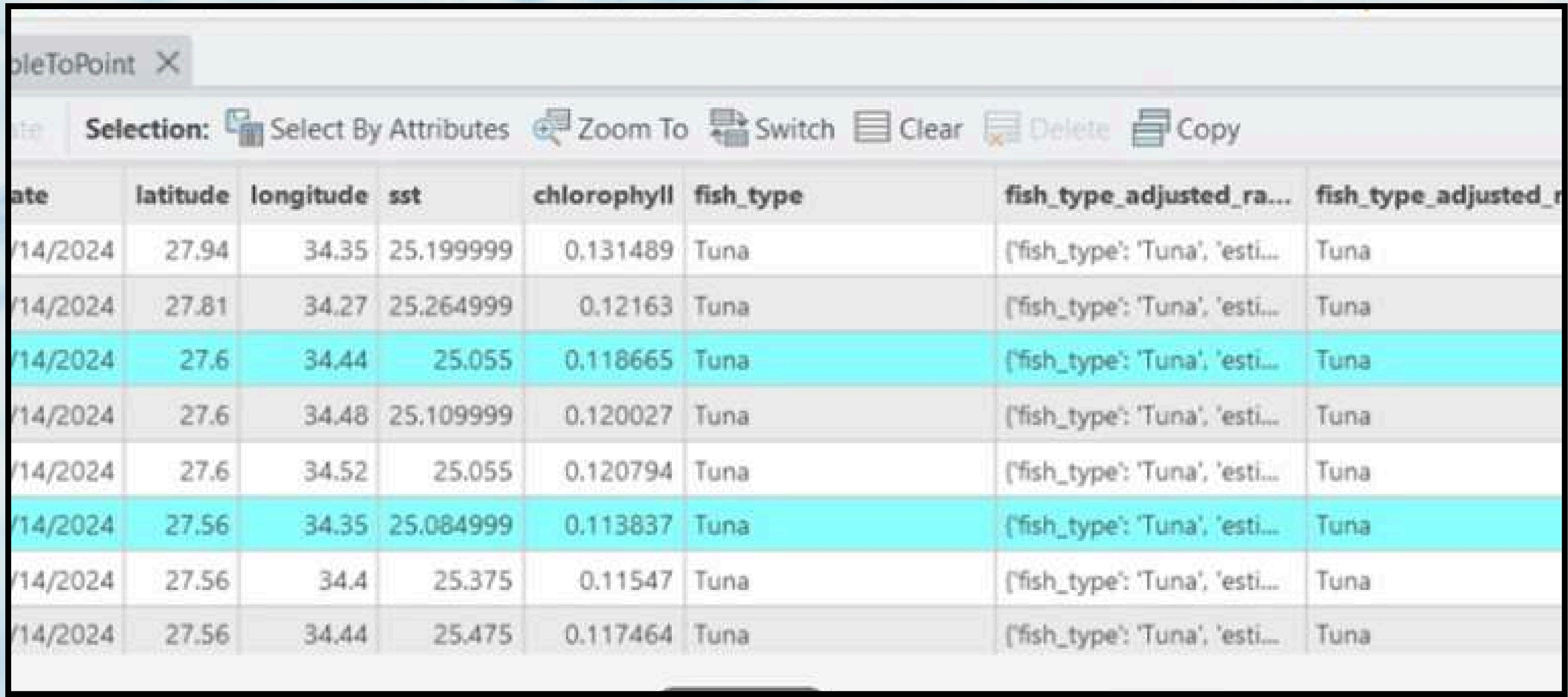


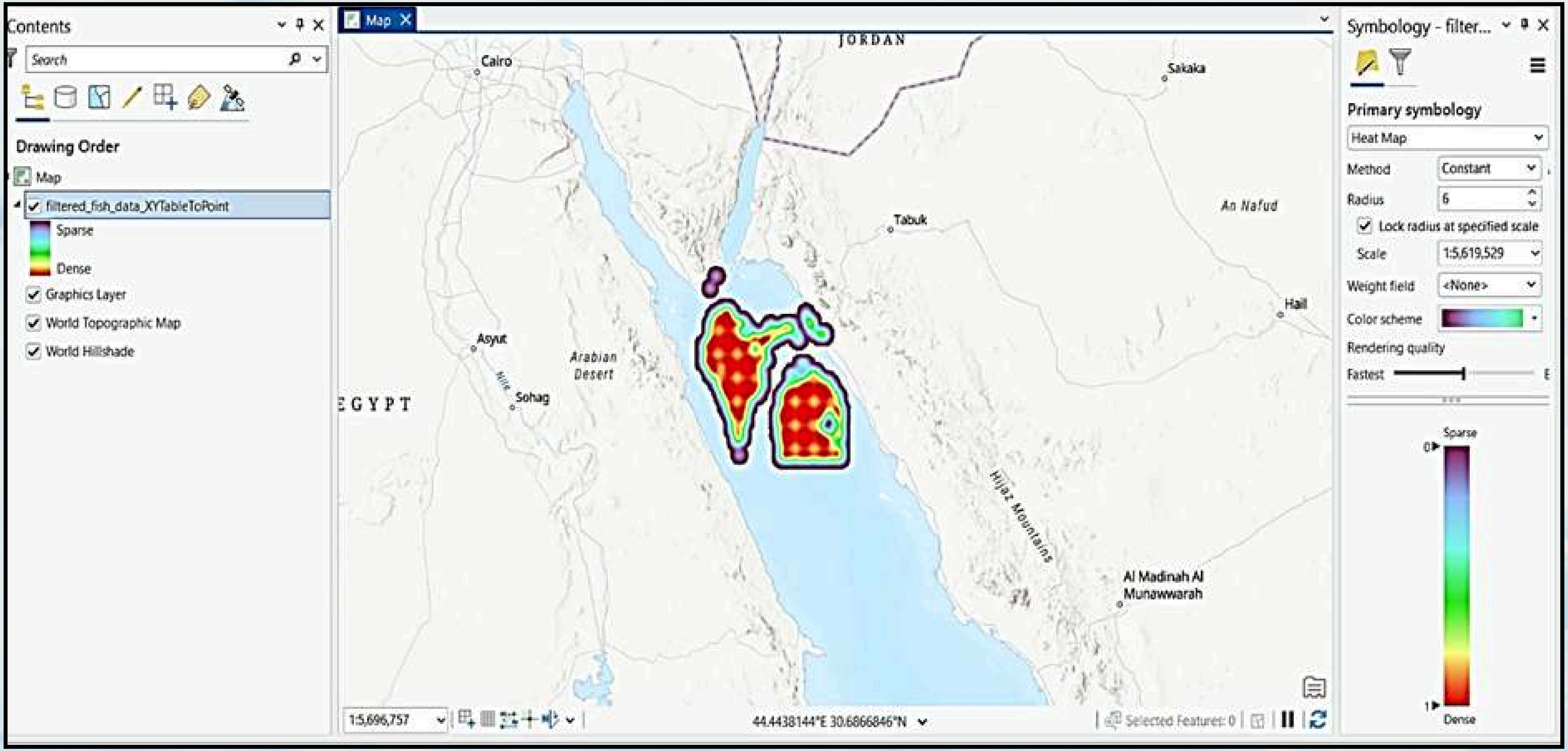
FIgure presents the same point layer with the attribute table visible, displaying environmental values such as SST and chlorophyll-a





ate	latitude	longitude	sst	chlorophyll	fish_type	fish_type_adjusted_ra...	fish_type_adjusted_r
/14/2024	27.94	34.35	25.199999	0.131489	Tuna	{'fish_type': 'Tuna', 'esti...	Tuna
/14/2024	27.81	34.27	25.264999	0.12163	Tuna	{'fish_type': 'Tuna', 'esti...	Tuna
/14/2024	27.6	34.44	25.055	0.118665	Tuna	{'fish_type': 'Tuna', 'esti...	Tuna
/14/2024	27.6	34.48	25.109999	0.120027	Tuna	{'fish_type': 'Tuna', 'esti...	Tuna
/14/2024	27.6	34.52	25.055	0.120794	Tuna	{'fish_type': 'Tuna', 'esti...	Tuna
/14/2024	27.56	34.35	25.084999	0.113837	Tuna	{'fish_type': 'Tuna', 'esti...	Tuna
/14/2024	27.56	34.4	25.375	0.11547	Tuna	{'fish_type': 'Tuna', 'esti...	Tuna
/14/2024	27.56	34.44	25.475	0.117464	Tuna	{'fish_type': 'Tuna', 'esti...	Tuna

Heatmap showing fish detection density; warmer areas indicate higher suitability



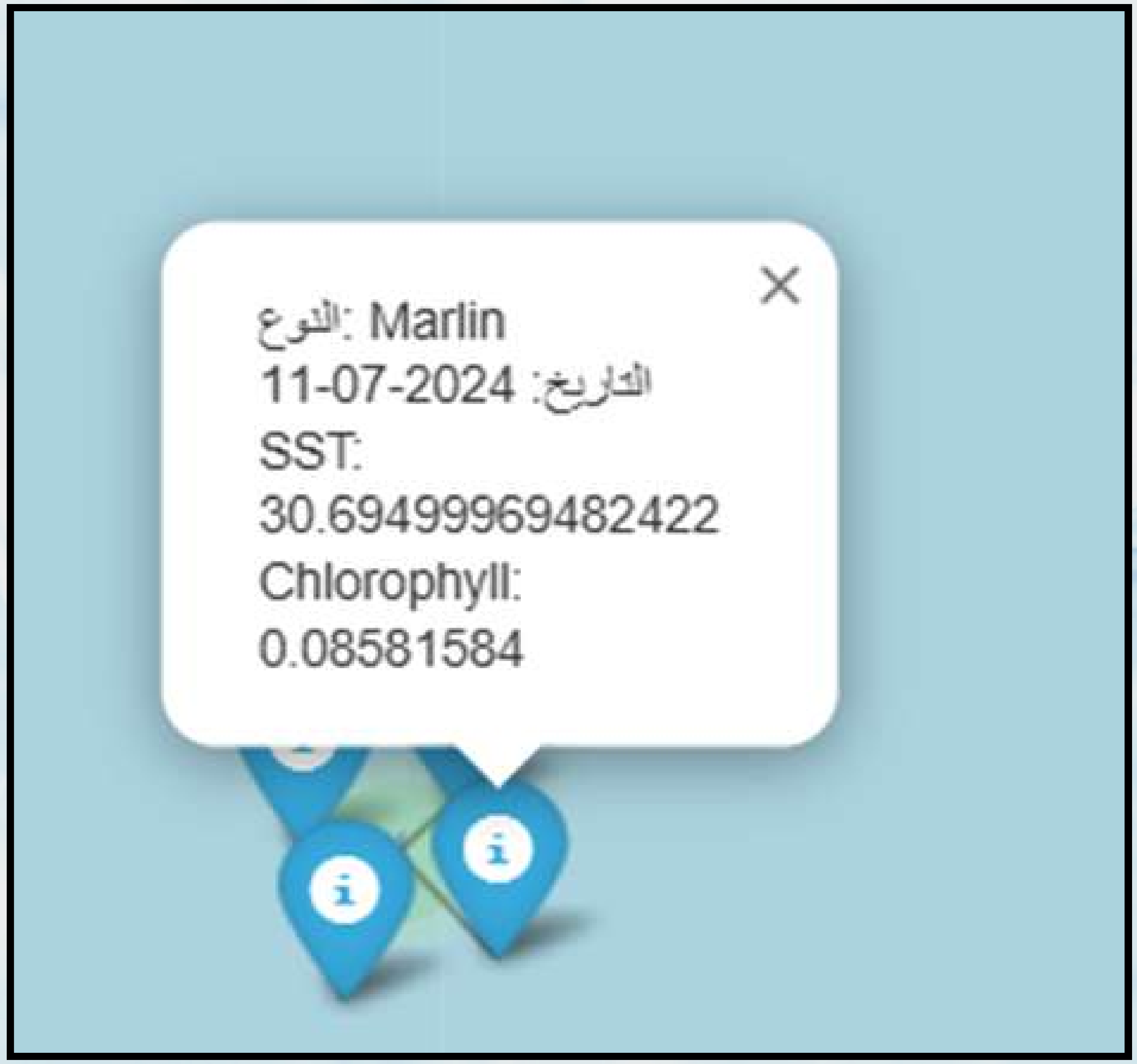
INTERACTIVE SYSTEM WITH PYTHON GUI

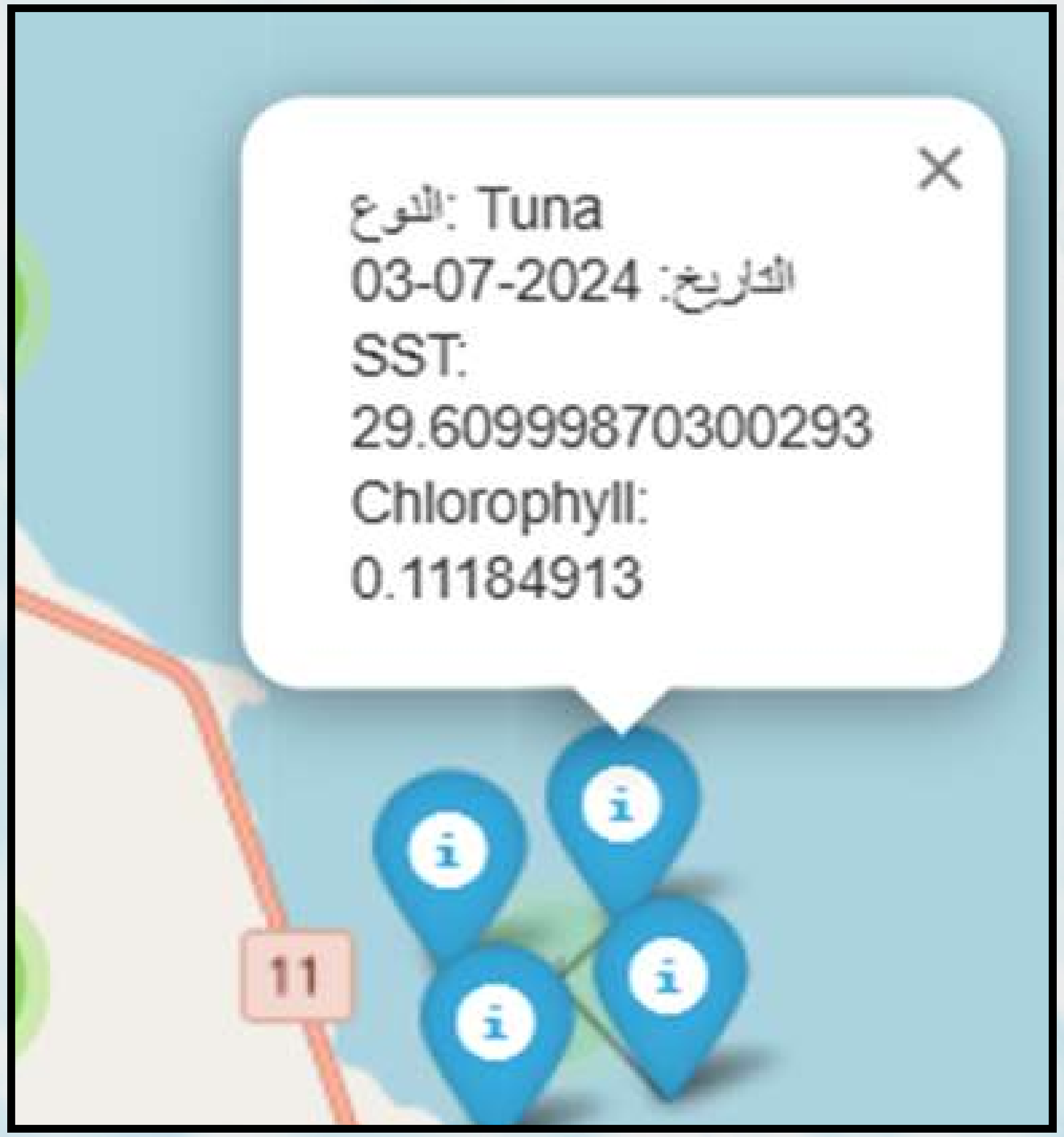
User enters the desired date into the GUI

Map appears showing fish locations for that date

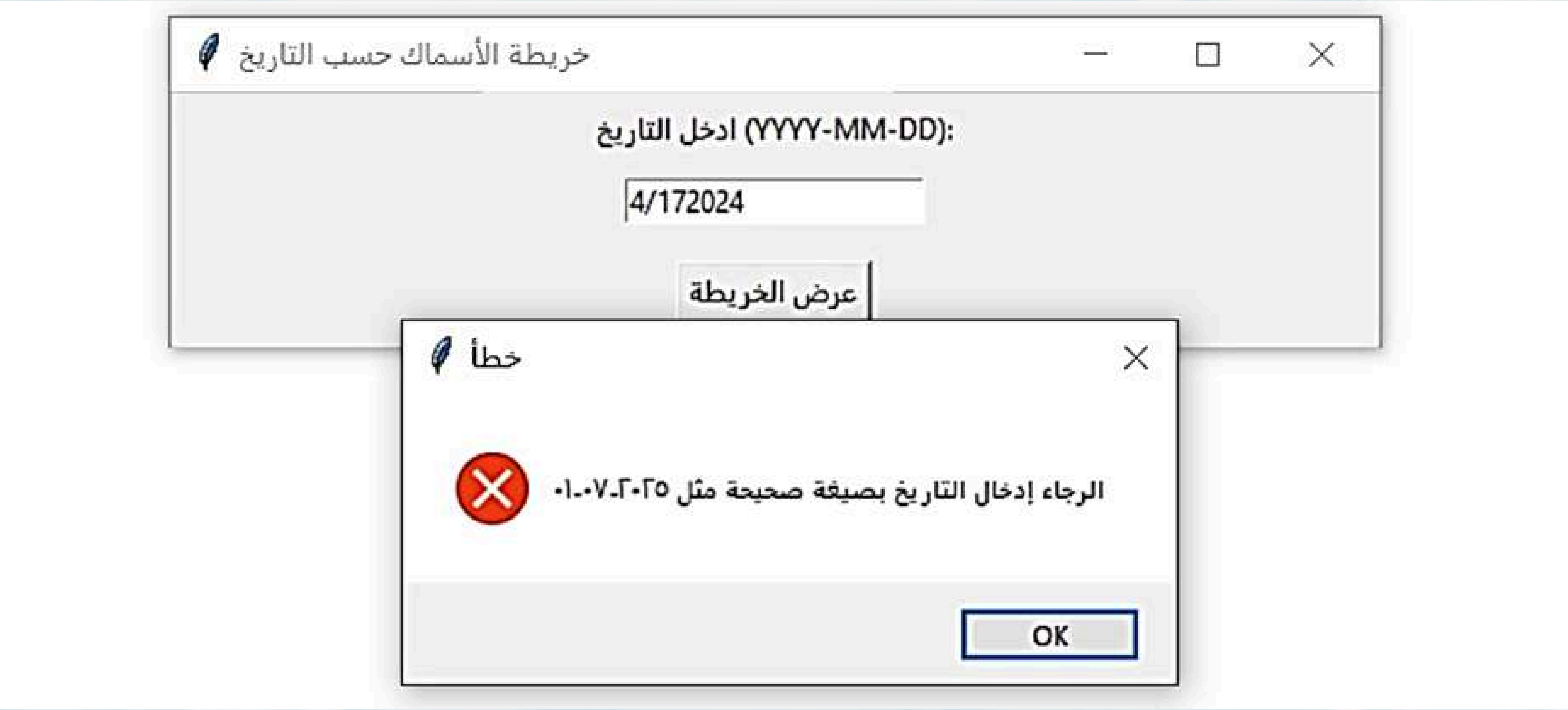
Clickable map markers reveal: fish type, SST, and chlorophyll

User clicks a location to view fish detection results with associated environmental data

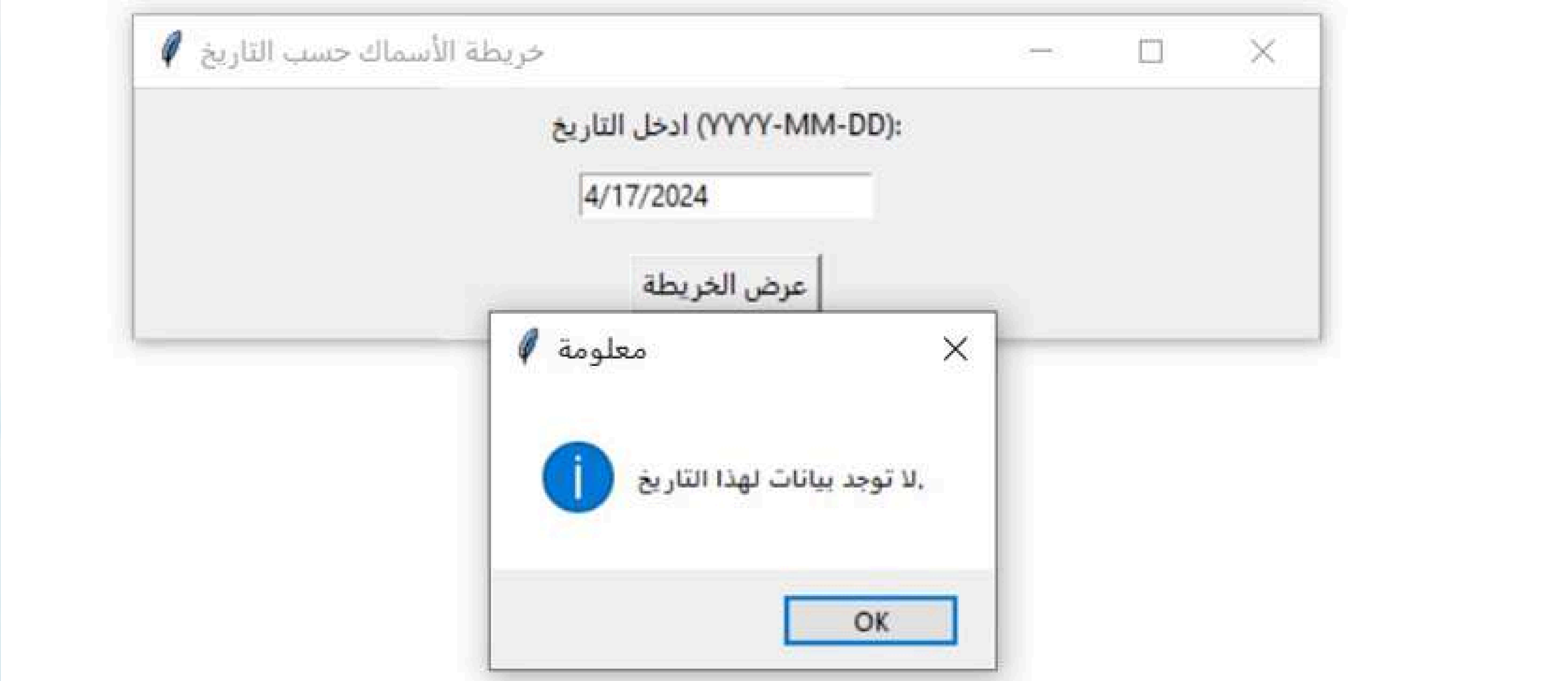




If the entered date is valid but no data records match it in the CSV file, the system notifies the user with an informational message



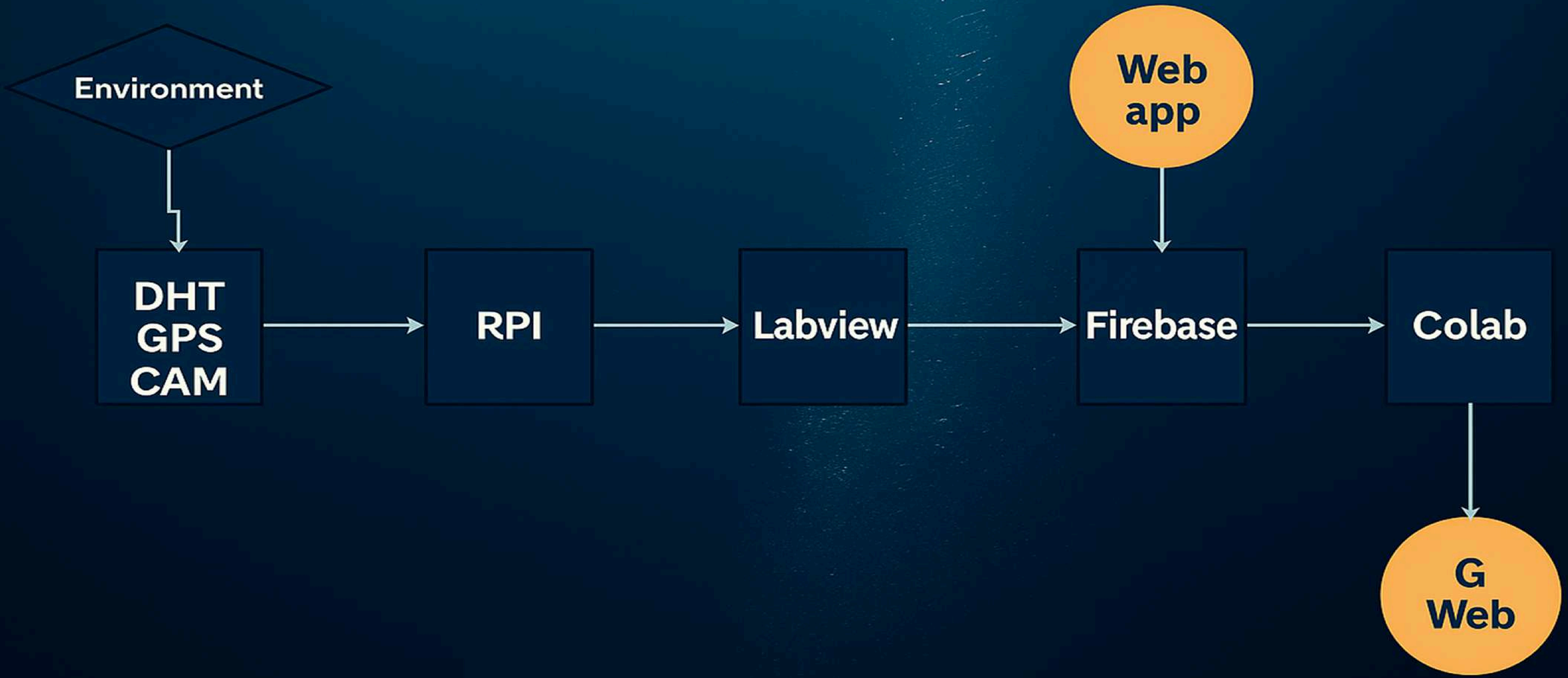
The message clearly states that no data is available for the selected date



WEB-BASED VISUALIZATION WITH FIREBASE INTEGRATION

System Interface

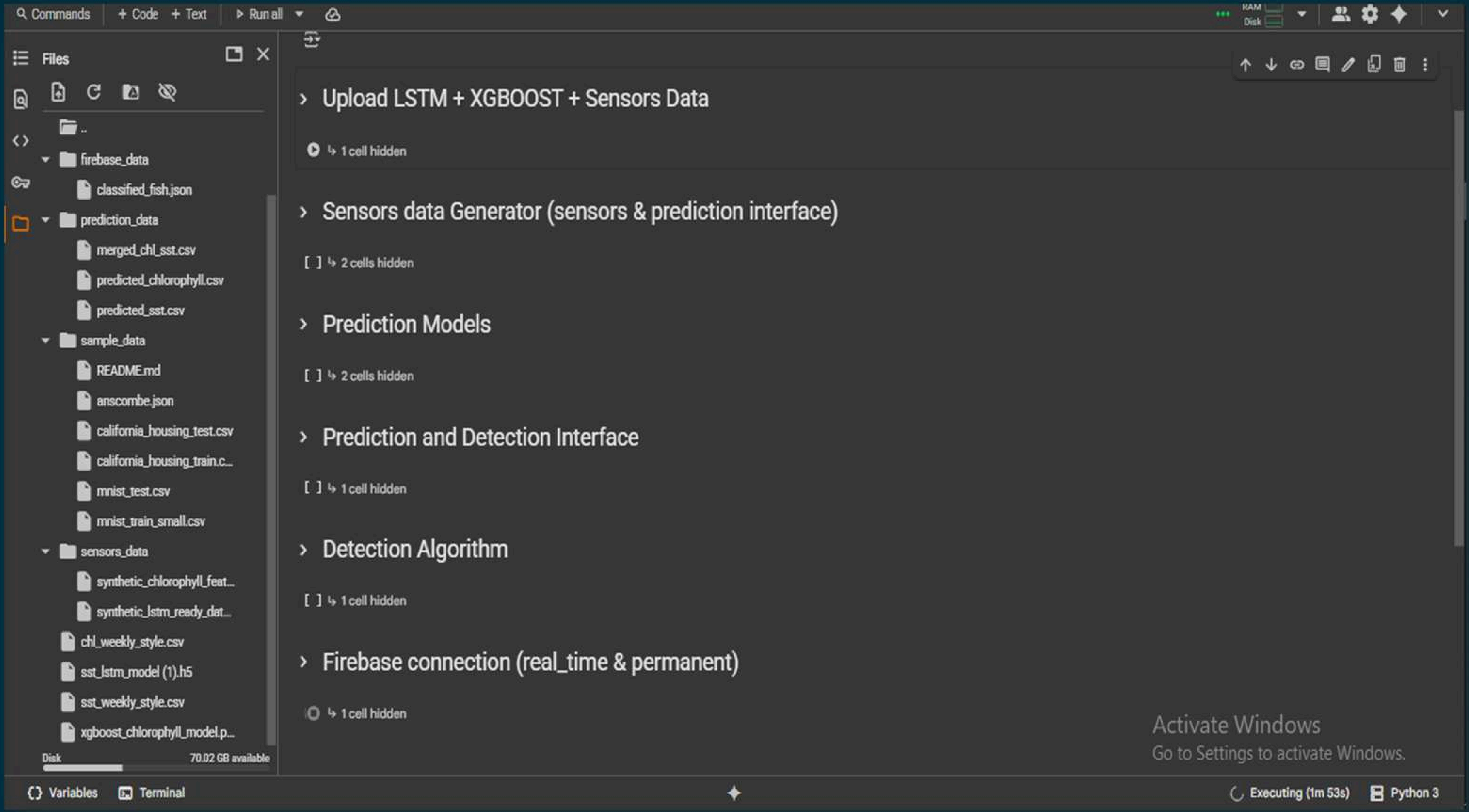
System Interface



2- Colab as the Core Interface

Colab acts as the main processing environment.

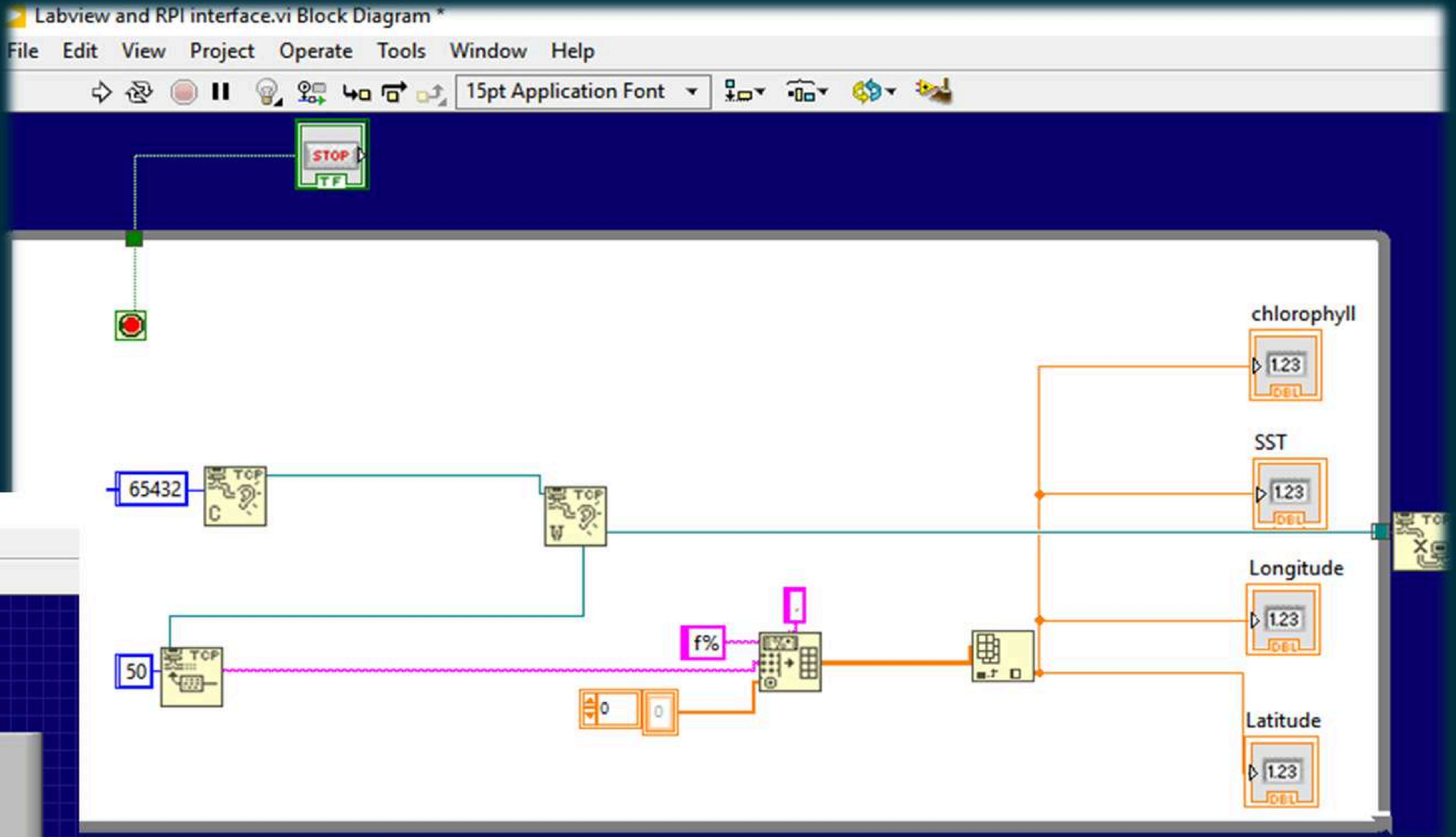
Handles data preprocessing, prediction, detection, and Firebase integration.

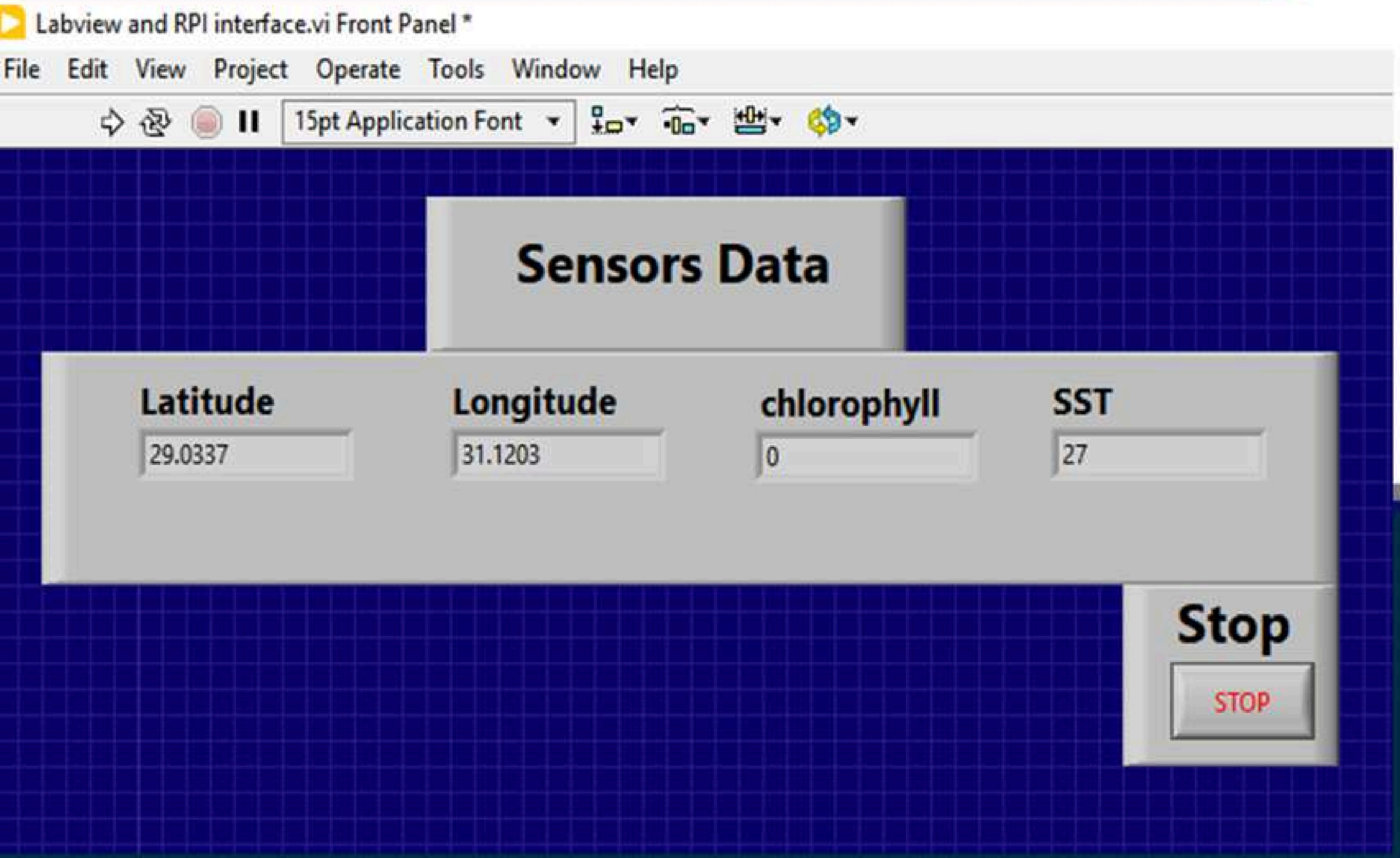


1-Raspberry Pi and LabVIEW Interface

Raspberry Pi reads from SST, Chlorophyll, and GPS

Raspberry Pi Sends data via Tcp/Ip to LabVIEW.





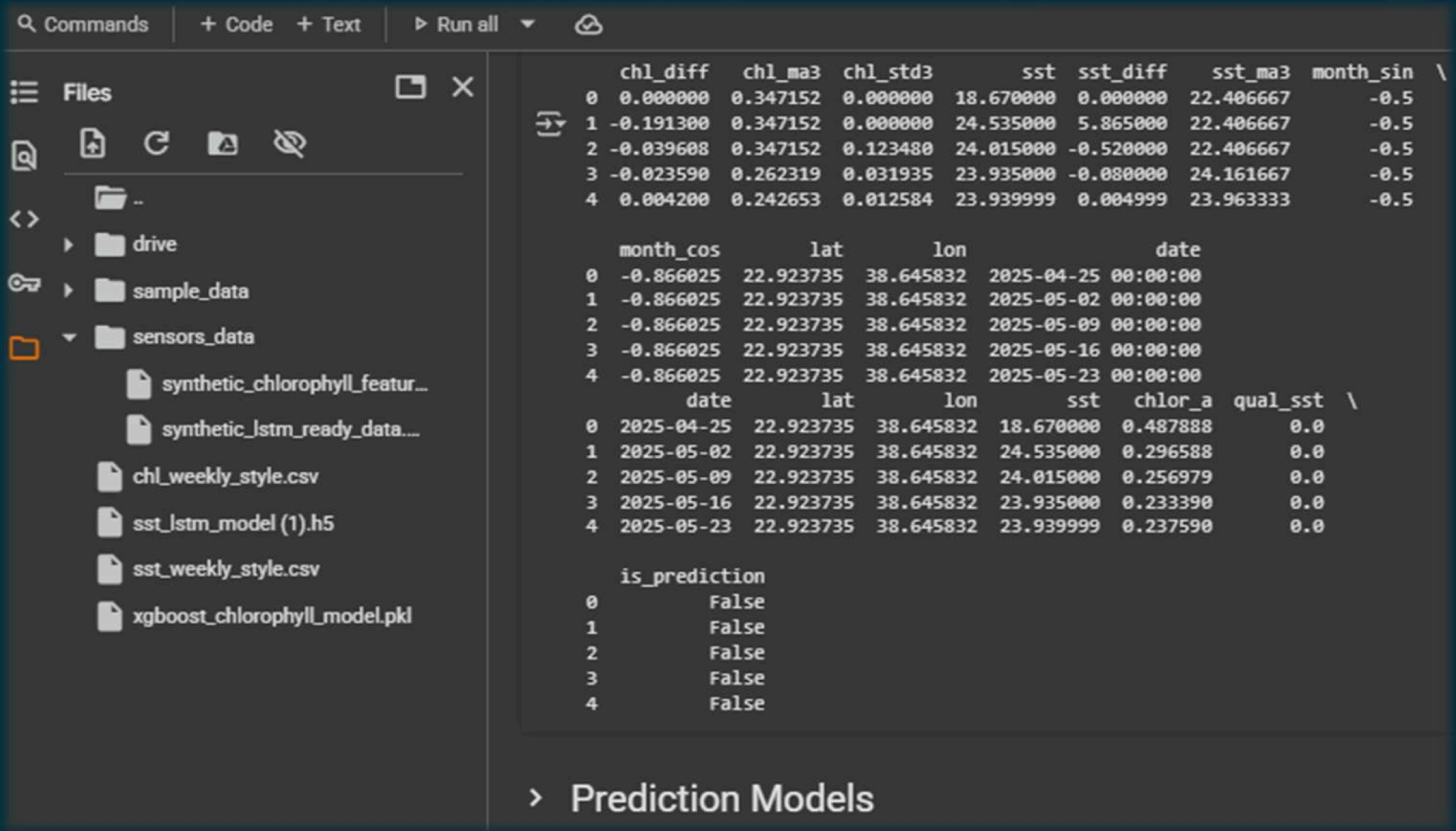
LabVIEW displays the data live and pushes it to Firebase.

3- Data Sensors & Prediction Models Interface

Simulated sensor data (CSV) is uploaded to Colab.

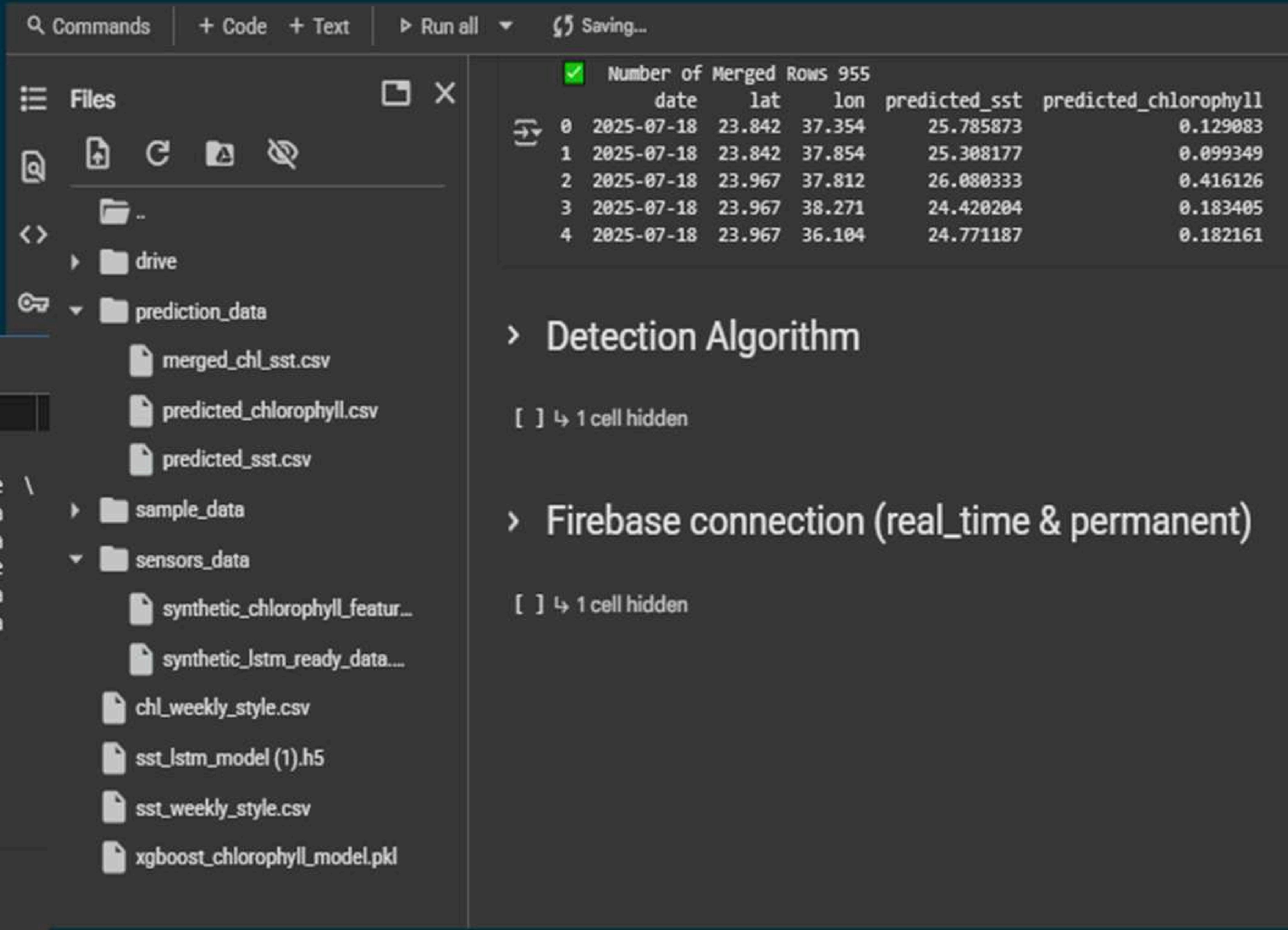
Preparing Two Dataframes for LSTM (SST) and XGBoost (Chlorophyll).

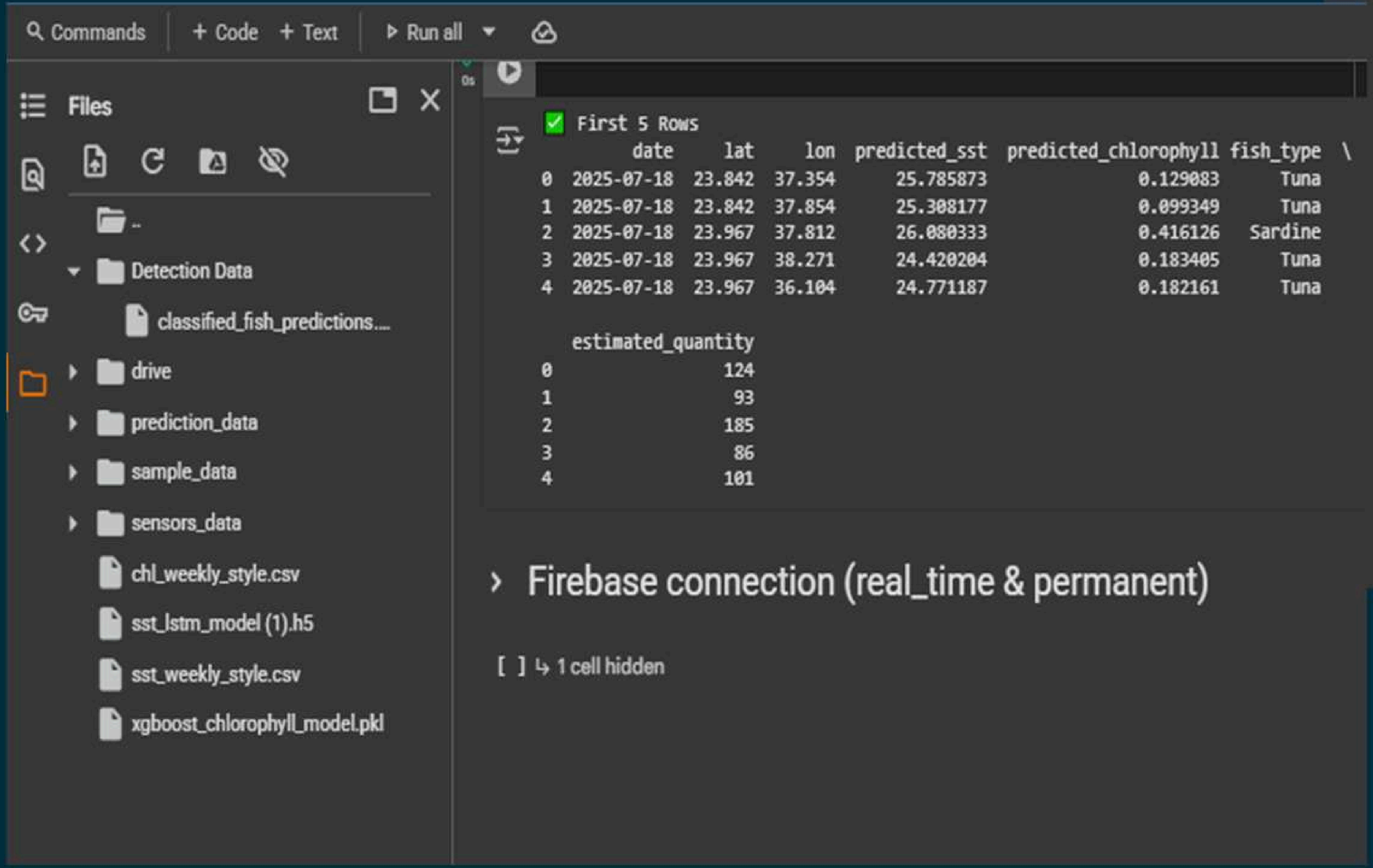
Outputs are saved into separate CSV files.



4- Prediction Data & Detection Fish Interface

Predicted SST and Chlorophyll are merged by lat/lon & date.

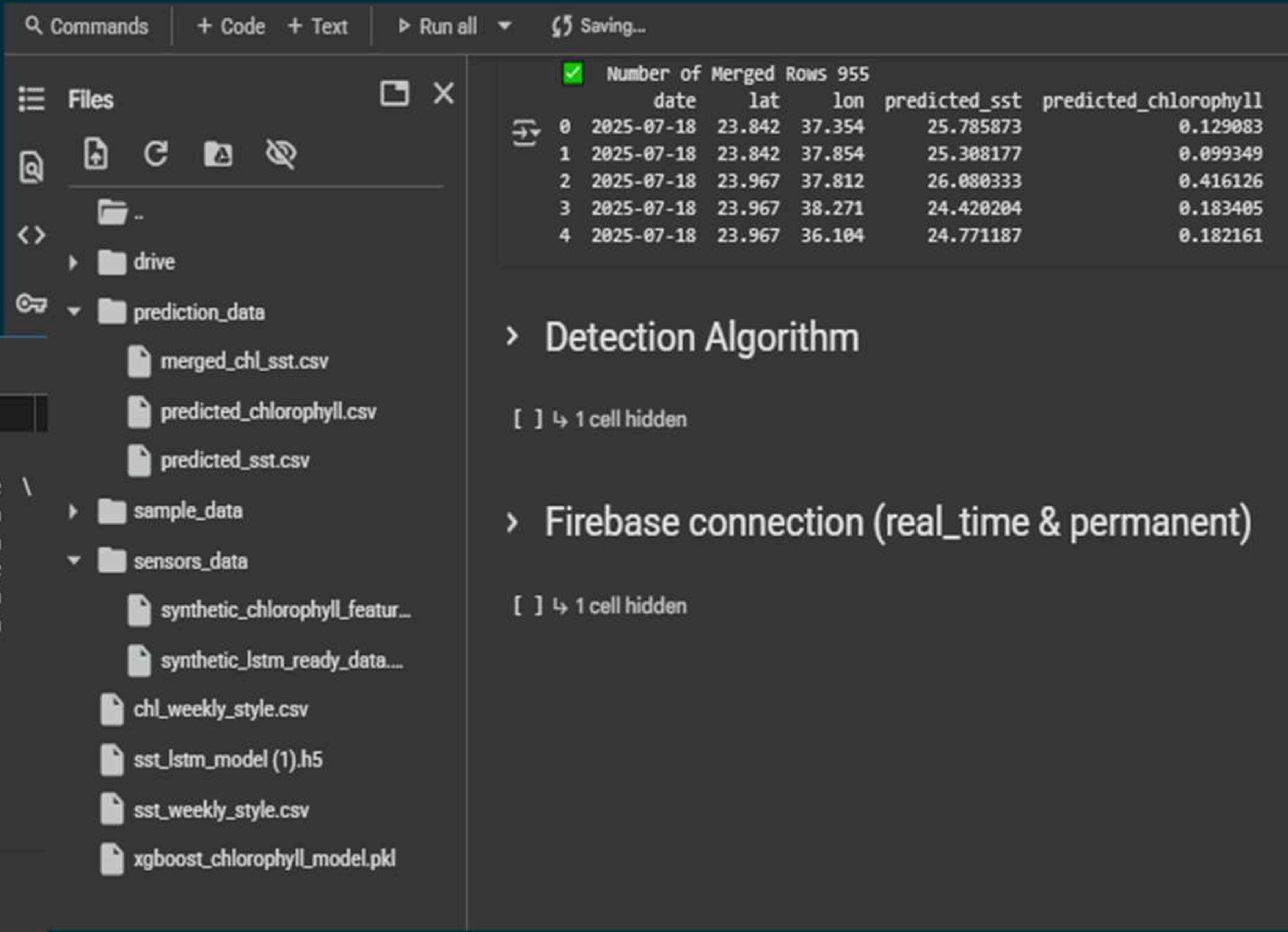


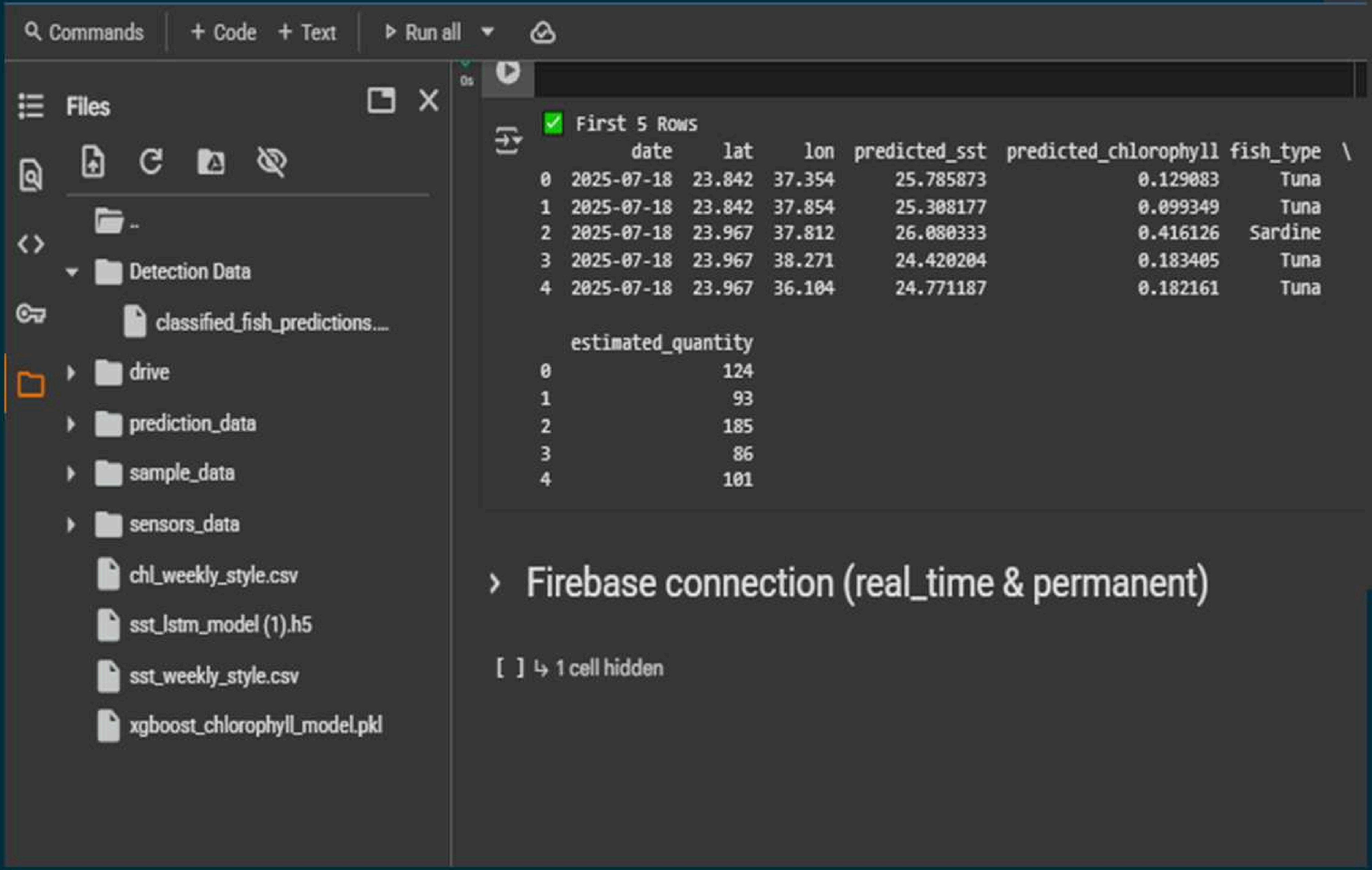


A fish classification algorithm is applied to estimate fish type & quantity.

4- Prediction Data & Detection Fish Interface

Predicted SST and Chlorophyll are merged by lat/lon & date.





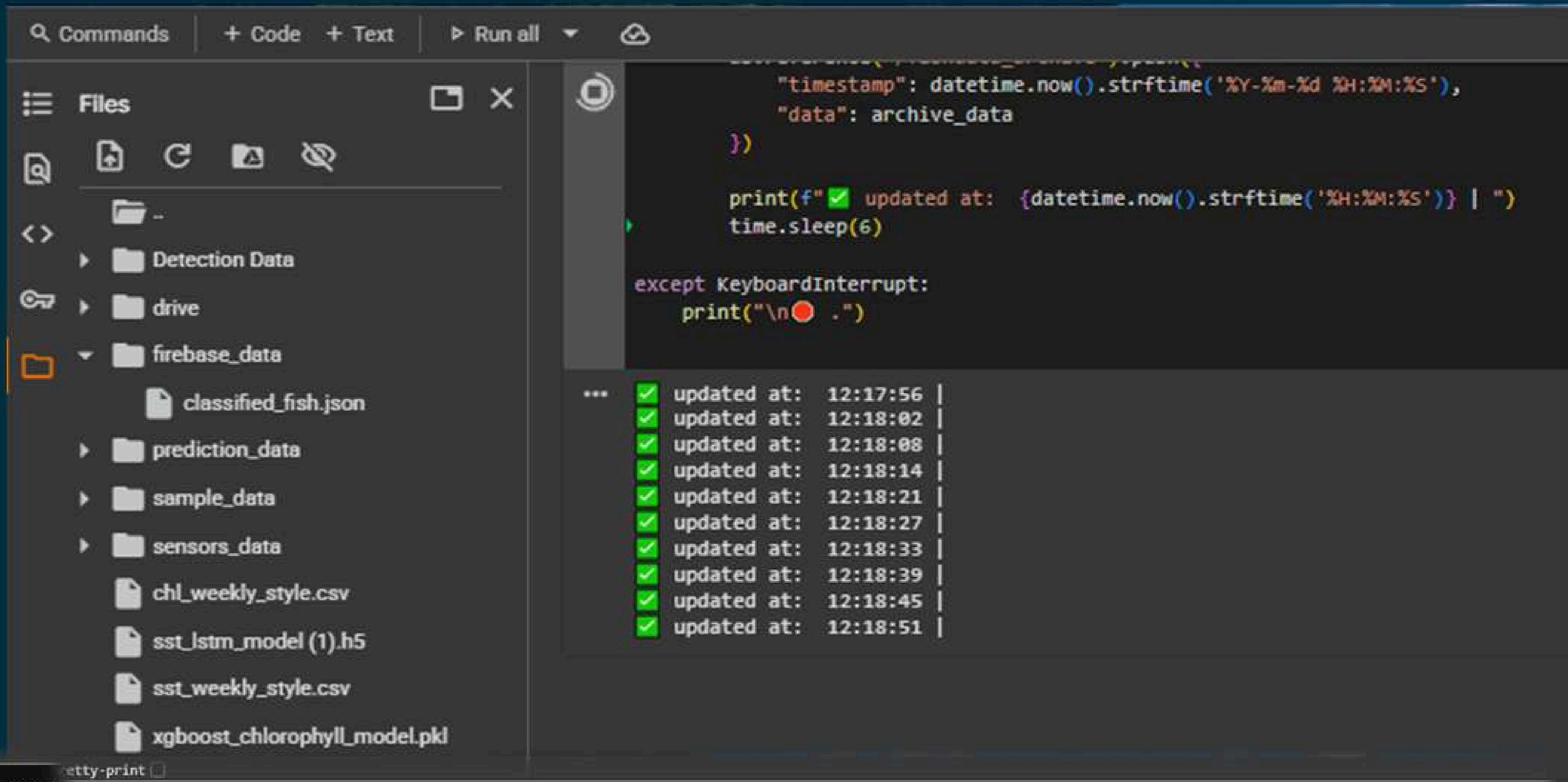
A fish classification algorithm is applied to estimate fish type & quantity.

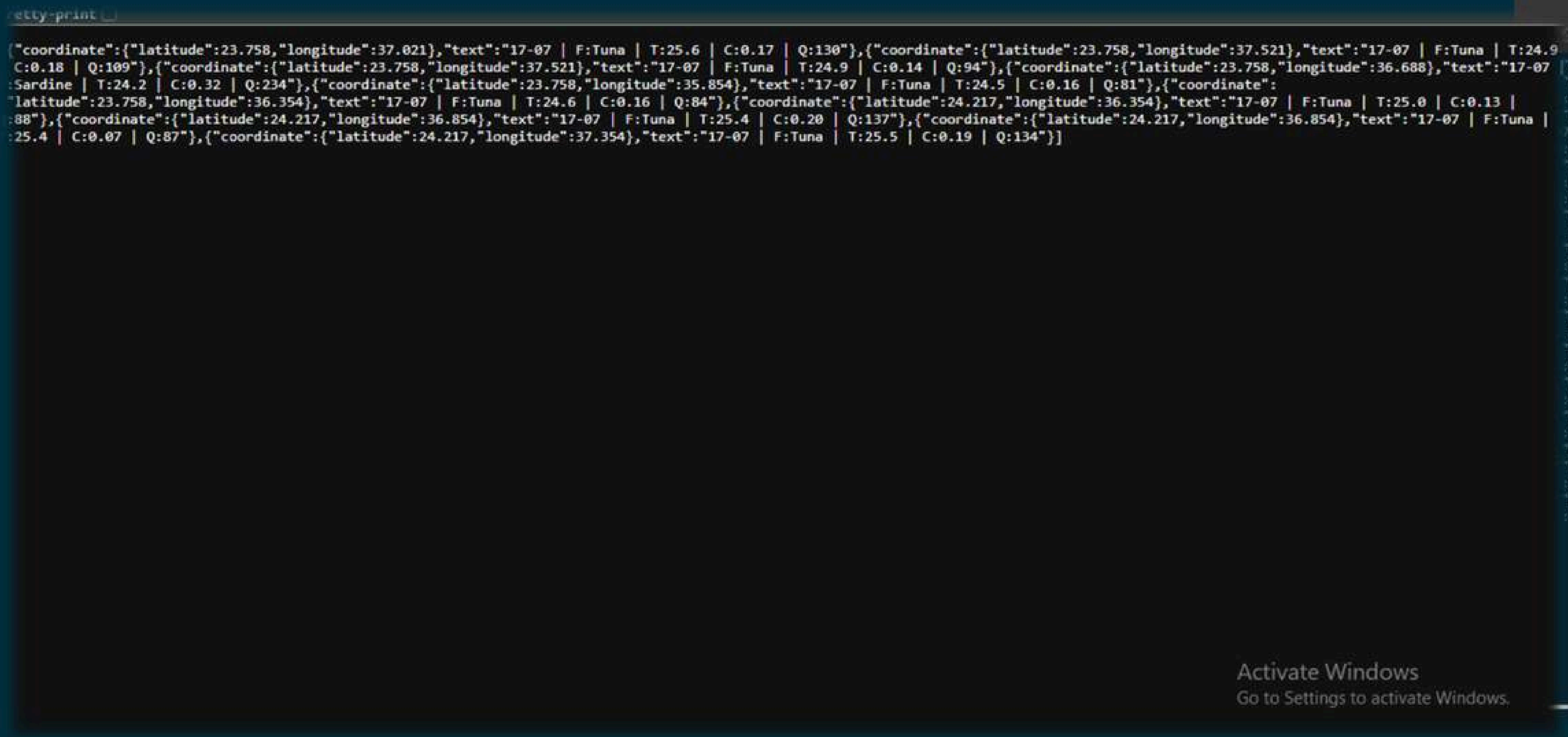
5- Firebase Upload Interface(Realtime & Archive)

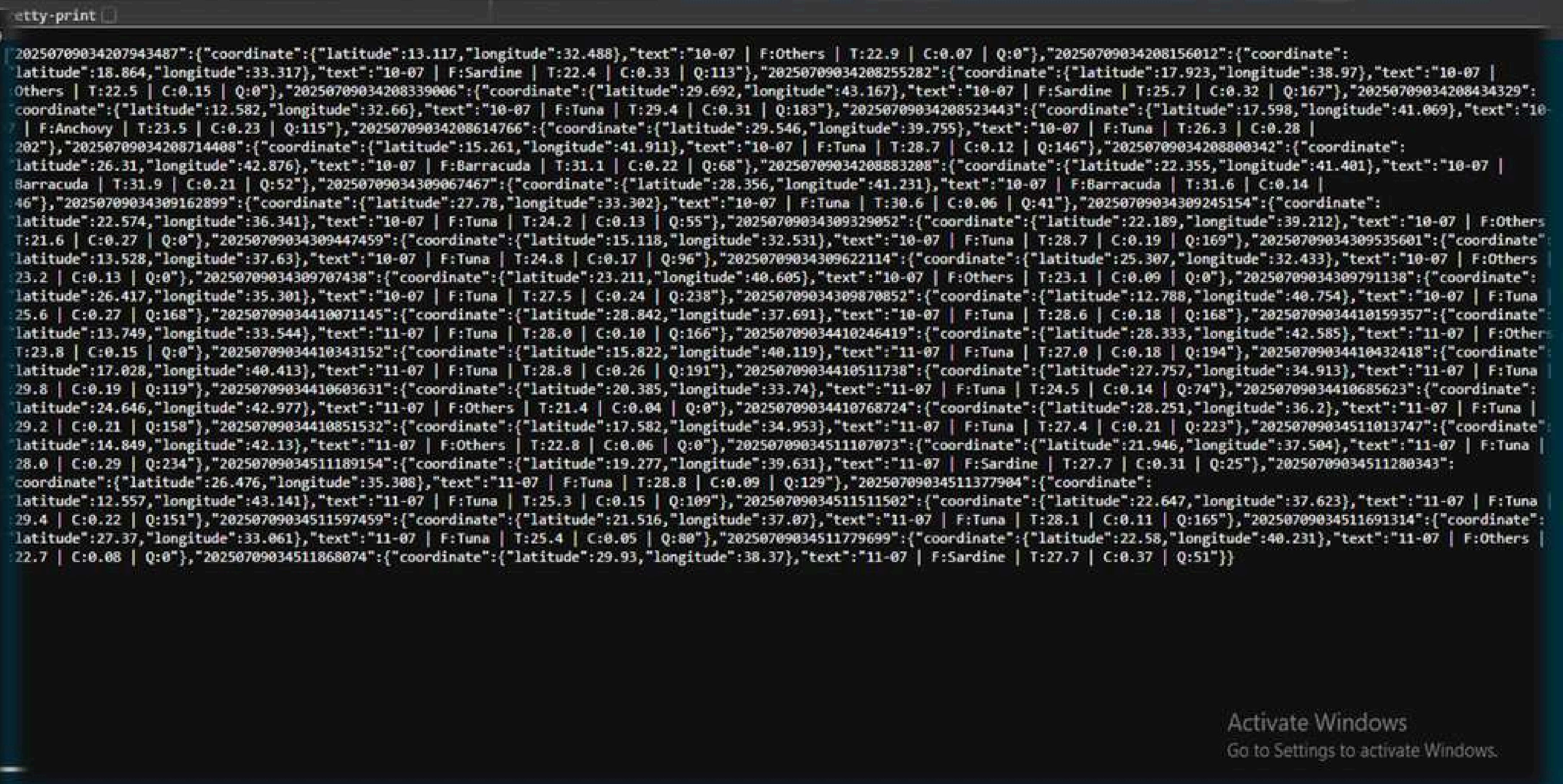
Final data is pushed to Firebase in
two paths:

/fishdata (json file): last 20 records (updated hourly)

/fishdata_archive (database): all data saved with timestamps



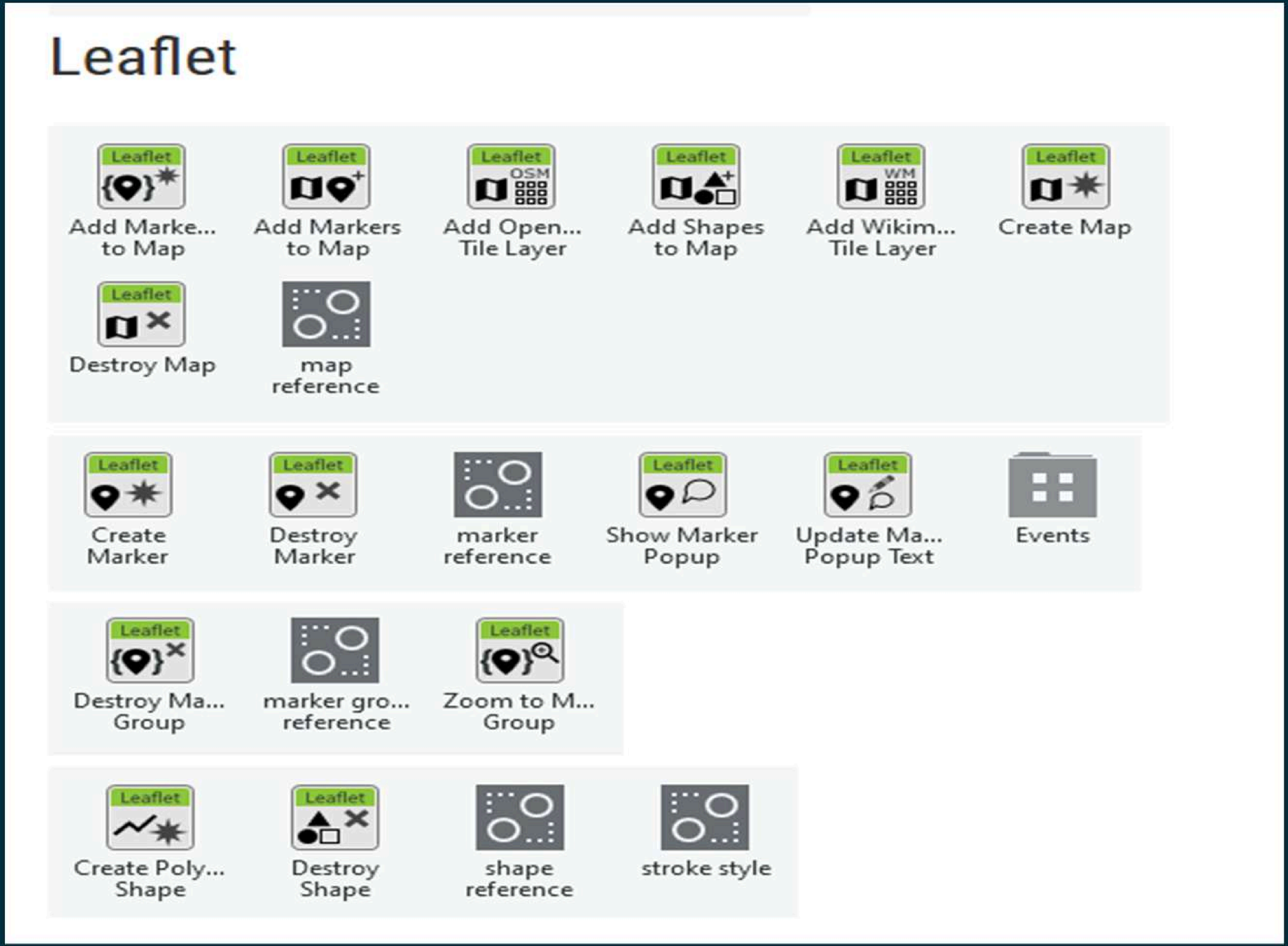




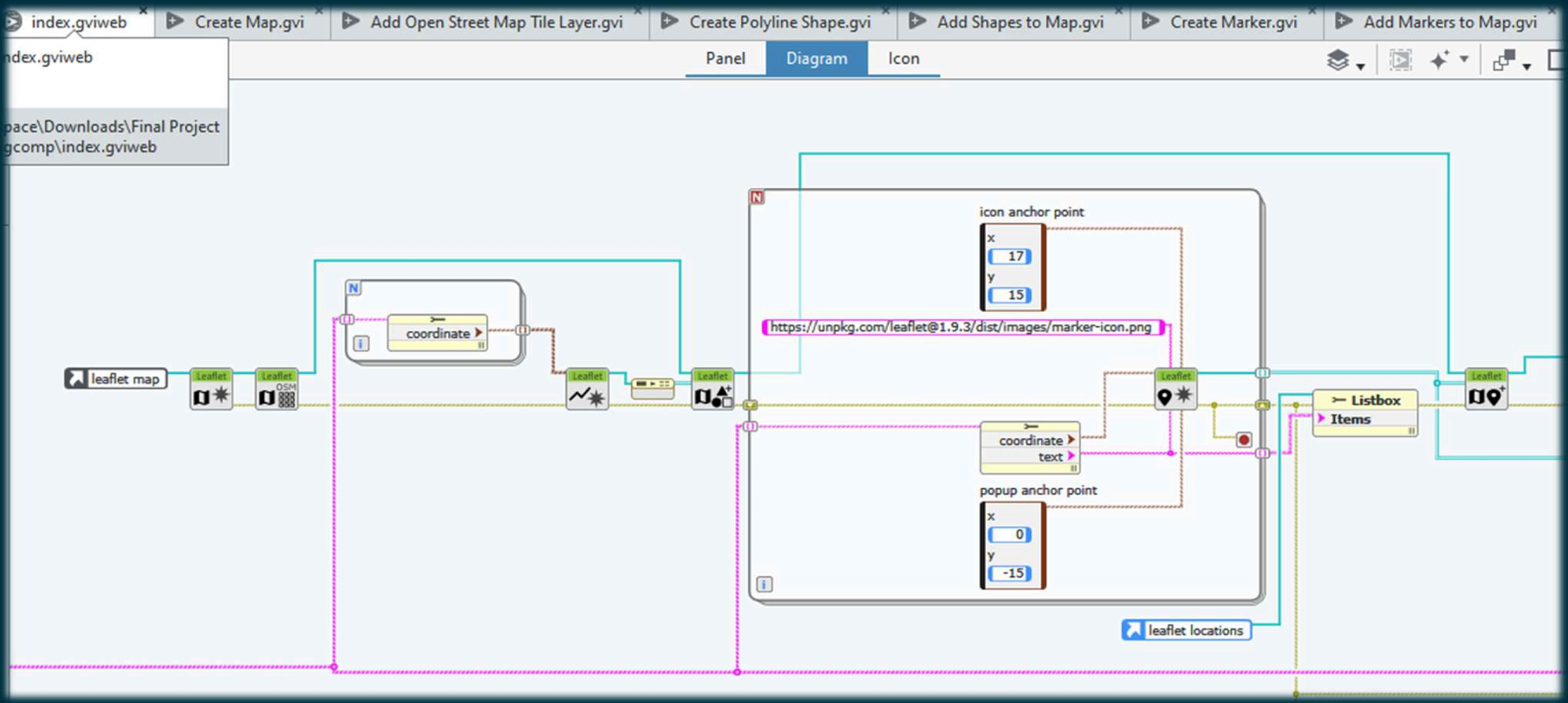
6- G Web Interface (Realtime display)

Leaflet Was Chosen It is open-source.

- ❑ Completely free to use.
- ❑ Easily integrates with G Web.
- ❑ Performs efficiently in browser-based interactive applications.
- ❑ Offers rich features such as marker clustering, popups, and shapes

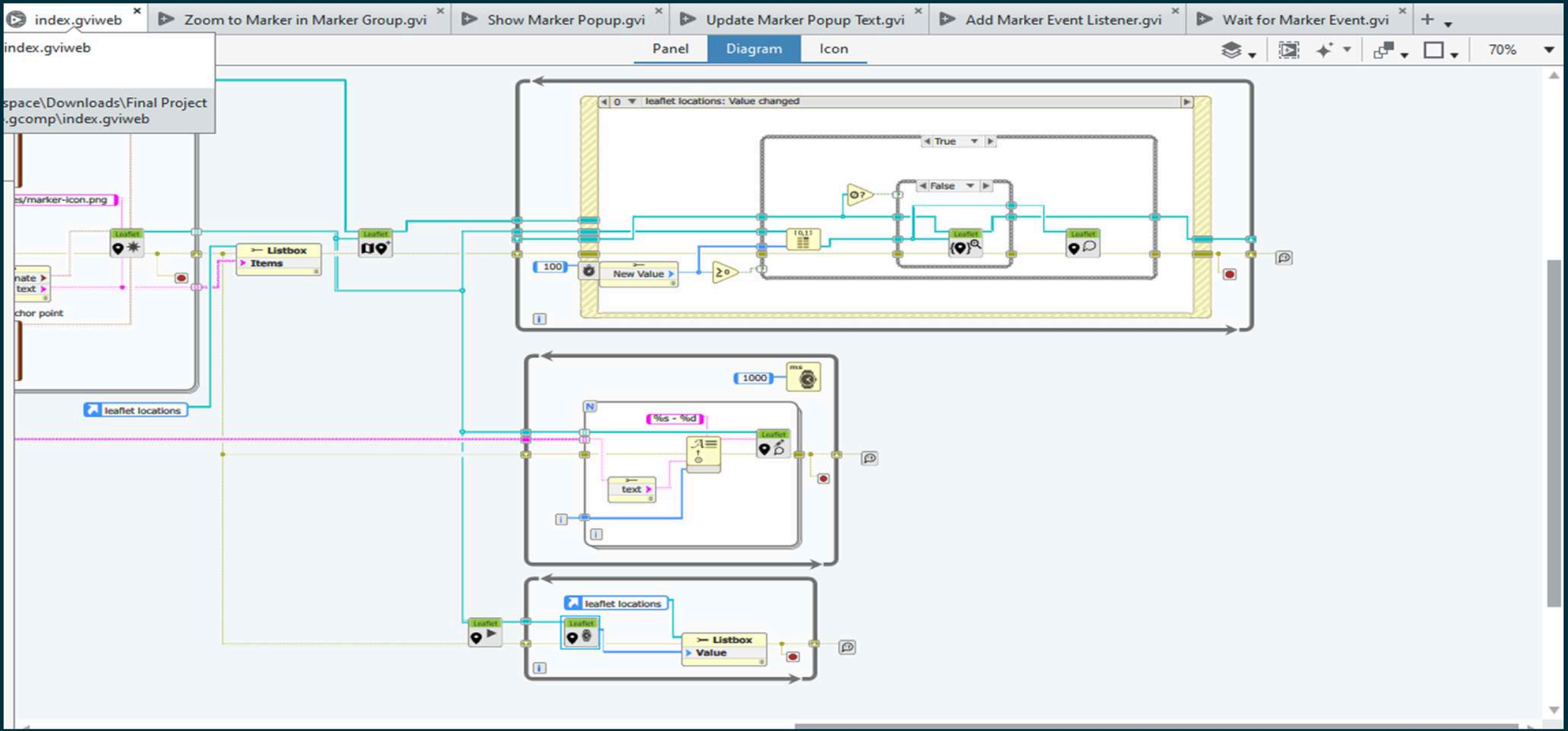


6- G Web Interface (Realtime display)



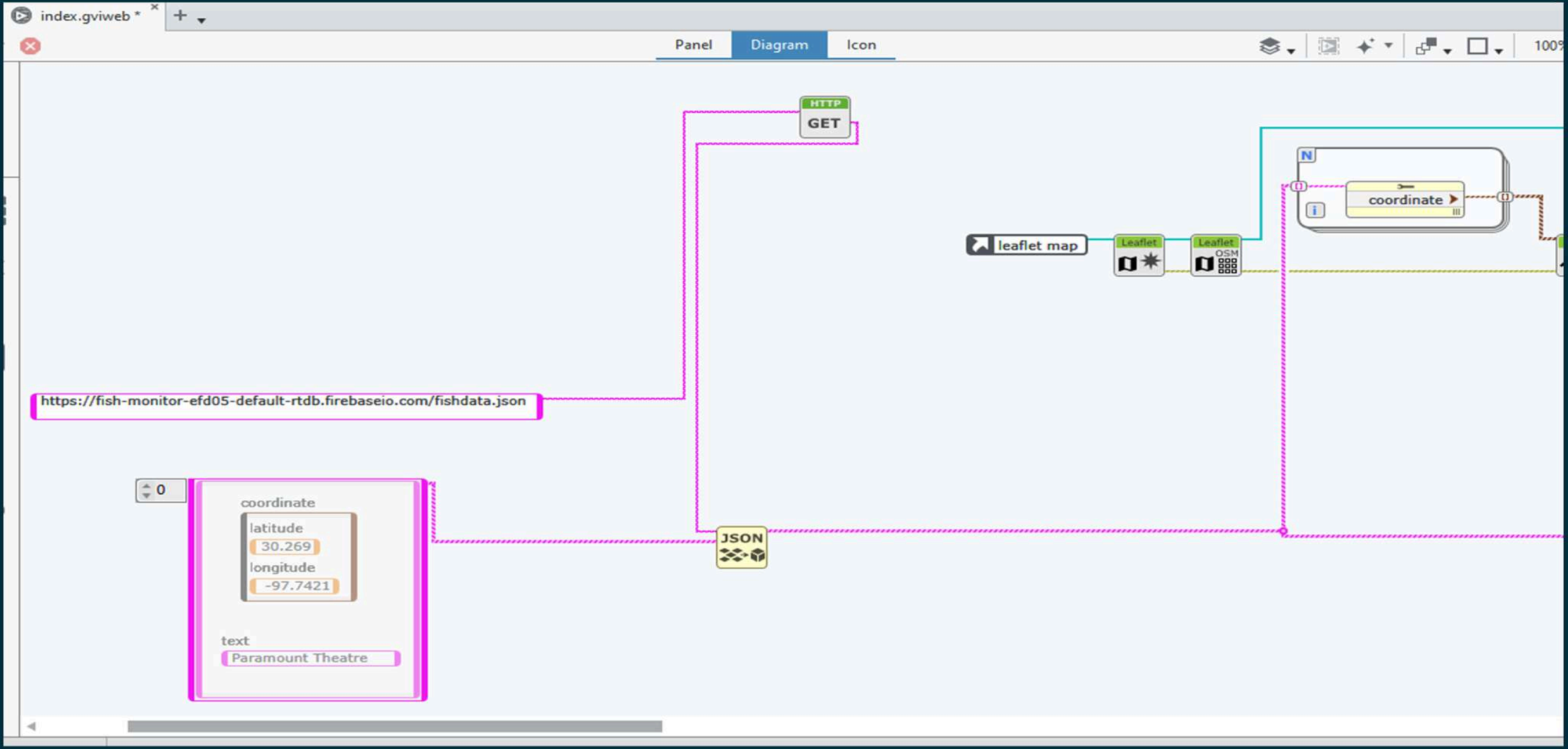
[create map] → [add map] → [create polyline and add it] → [create marker and add it]

6- G Web Interface (Realtime display)



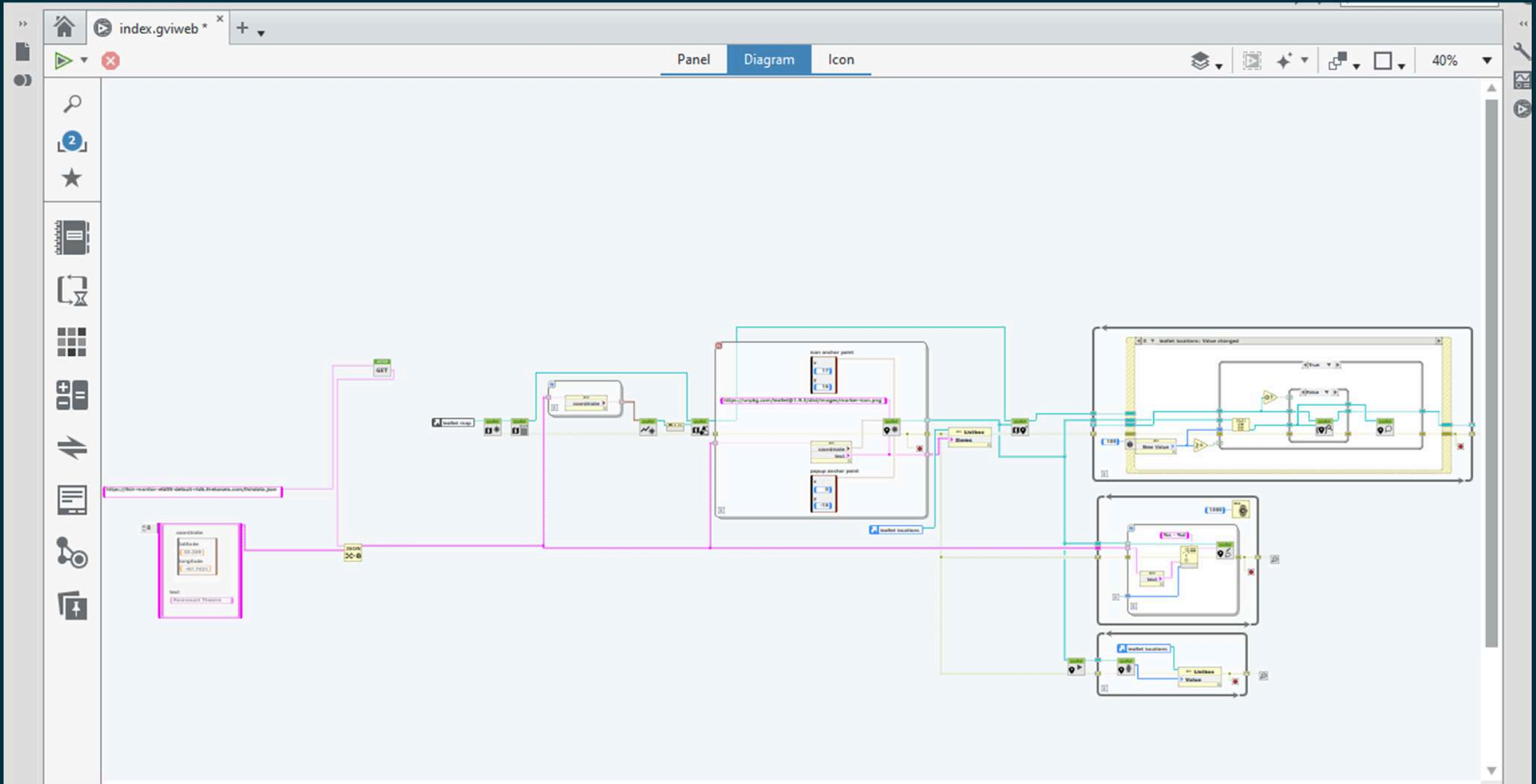
[Zoom to marker → marker popup] → [update popup] → [marker event listener → and wait]

6- G Web Interface (Realtime display)

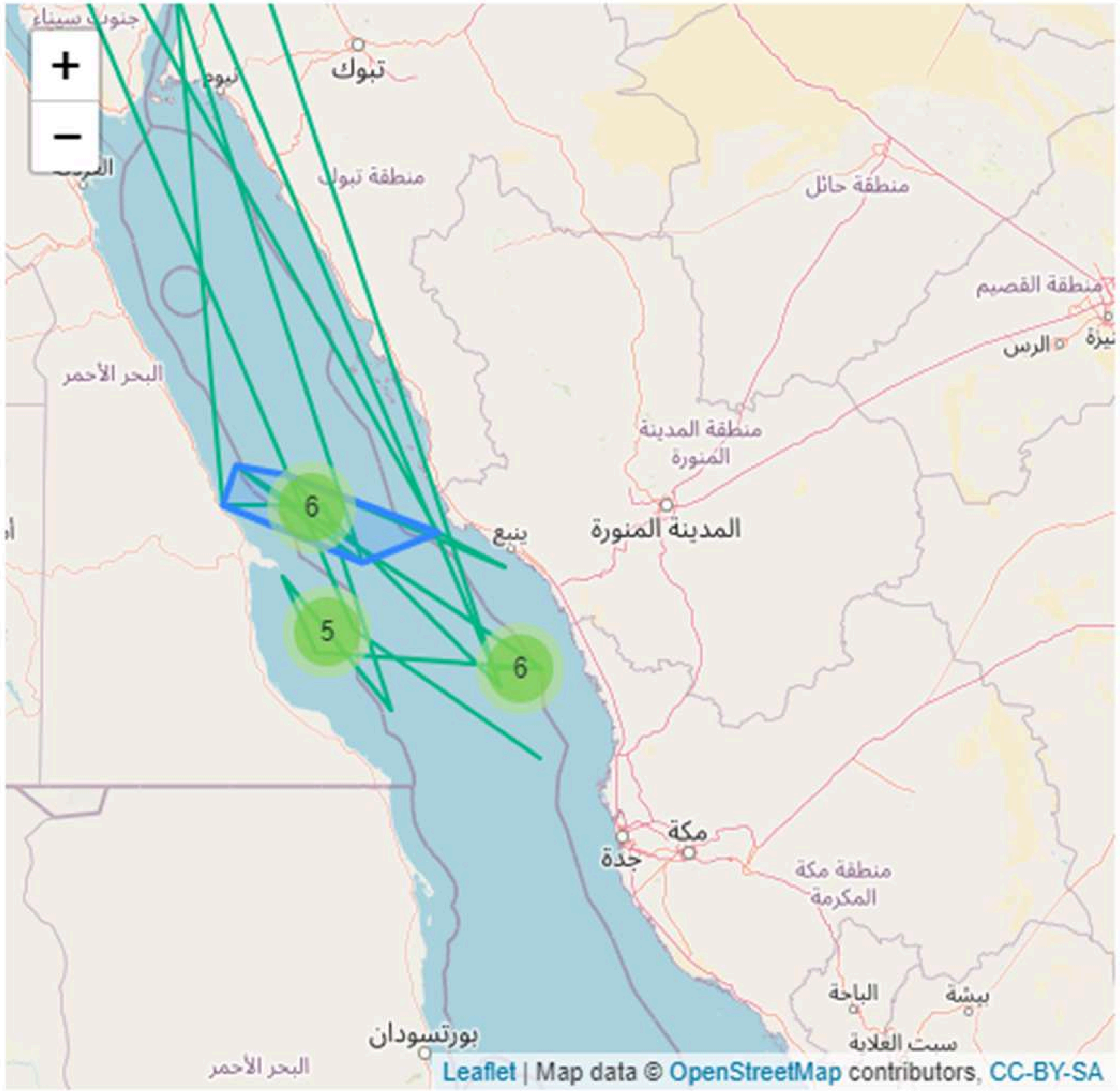


[Get Http] → [unflatten from Json] → [array cluster] → [Map Code]

6- G Web Interface (Realtime display)



Leaflet Map (Open Street Maps)



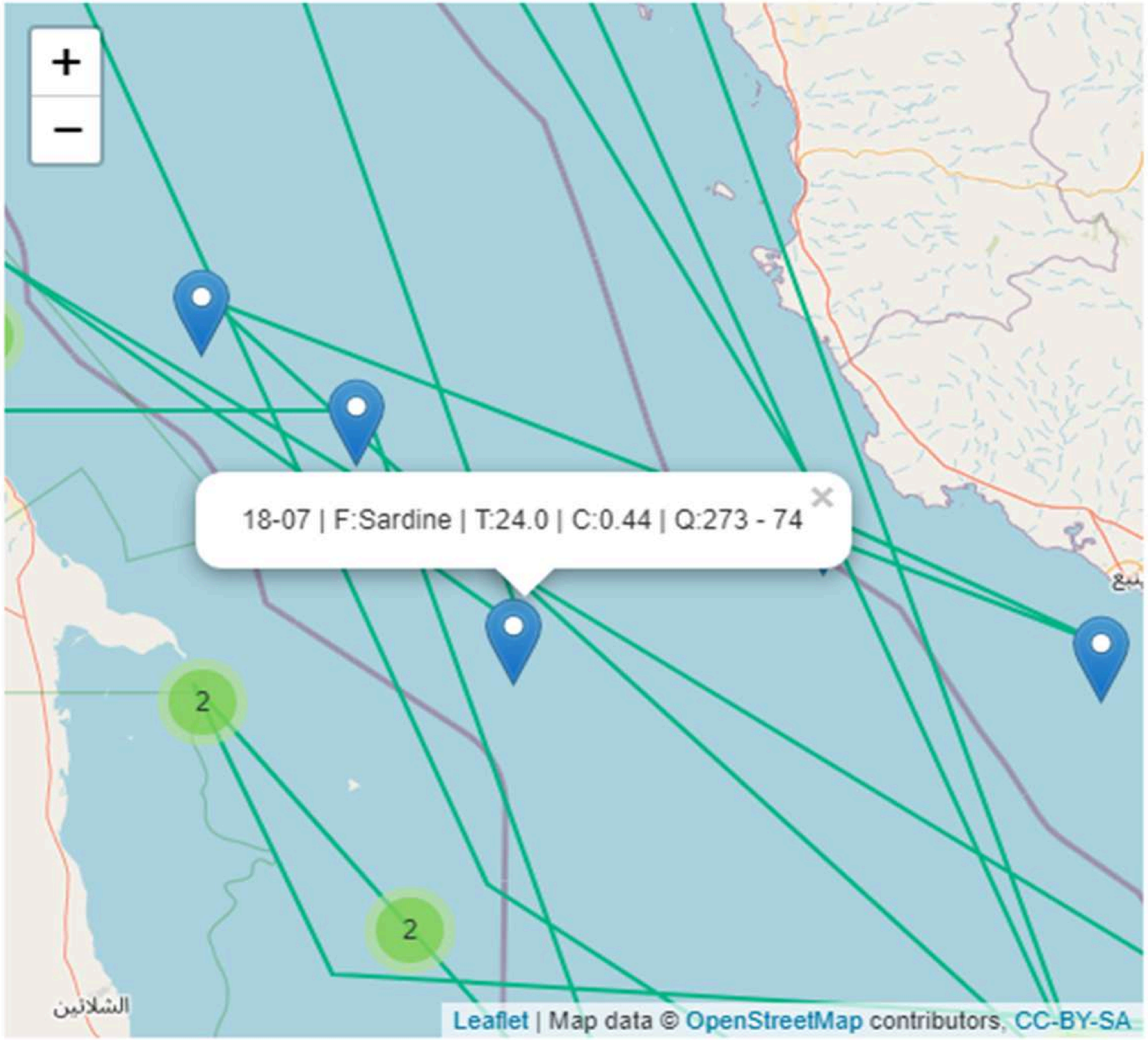
leaflet locations

18-07 | F:Sardine | T:25.6 | ...
18-07 | F:Tuna | T:24.5 | C:0....
18-07 | F:Sardine | T:24.0 | ...
18-07 | F:Tuna | T:25.3 | C:0....
18-07 | F:Tuna | T:26.2 | C:0....
18-07 | F:Sardine | T:24.2 | ...
18-07 | F:Tuna | T:25.0 | C:0....
18-07 | F:Tuna | T:24.5 | C:0....
18-07 | F:Others | T:24.0 | C:...
18-07 | F:Tuna | T:24.9 | C:0....
18-07 | F:Tuna | T:24.3 | C:0....
18-07 | F:Tuna | T:25.1 | C:0....

leaflet markers

individual grouped

Leaflet Map (Open Street Maps)



leaflet locations

- 18-07 | F:Sardine | T:25.6 | ...
- 18-07 | F:Tuna | T:24.5 | C:0....
- 18-07 | F:Sardine | T:24.0 | ...
- 18-07 | F:Tuna | T:25.3 | C:0....
- 18-07 | F:Tuna | T:26.2 | C:0....
- 18-07 | F:Sardine | T:24.2 | ...
- 18-07 | F:Tuna | T:25.0 | C:0....
- 18-07 | F:Tuna | T:24.5 | C:0....
- 18-07 | F:Others | T:24.0 | C:...
- 18-07 | F:Tuna | T:24.9 | C:0....
- 18-07 | F:Tuna | T:24.3 | C:0....
- 18-07 | F:Tuna | T:25.1 | C:0....

leaflet markers

individual grouped

The Structure of cubesat

Introduction

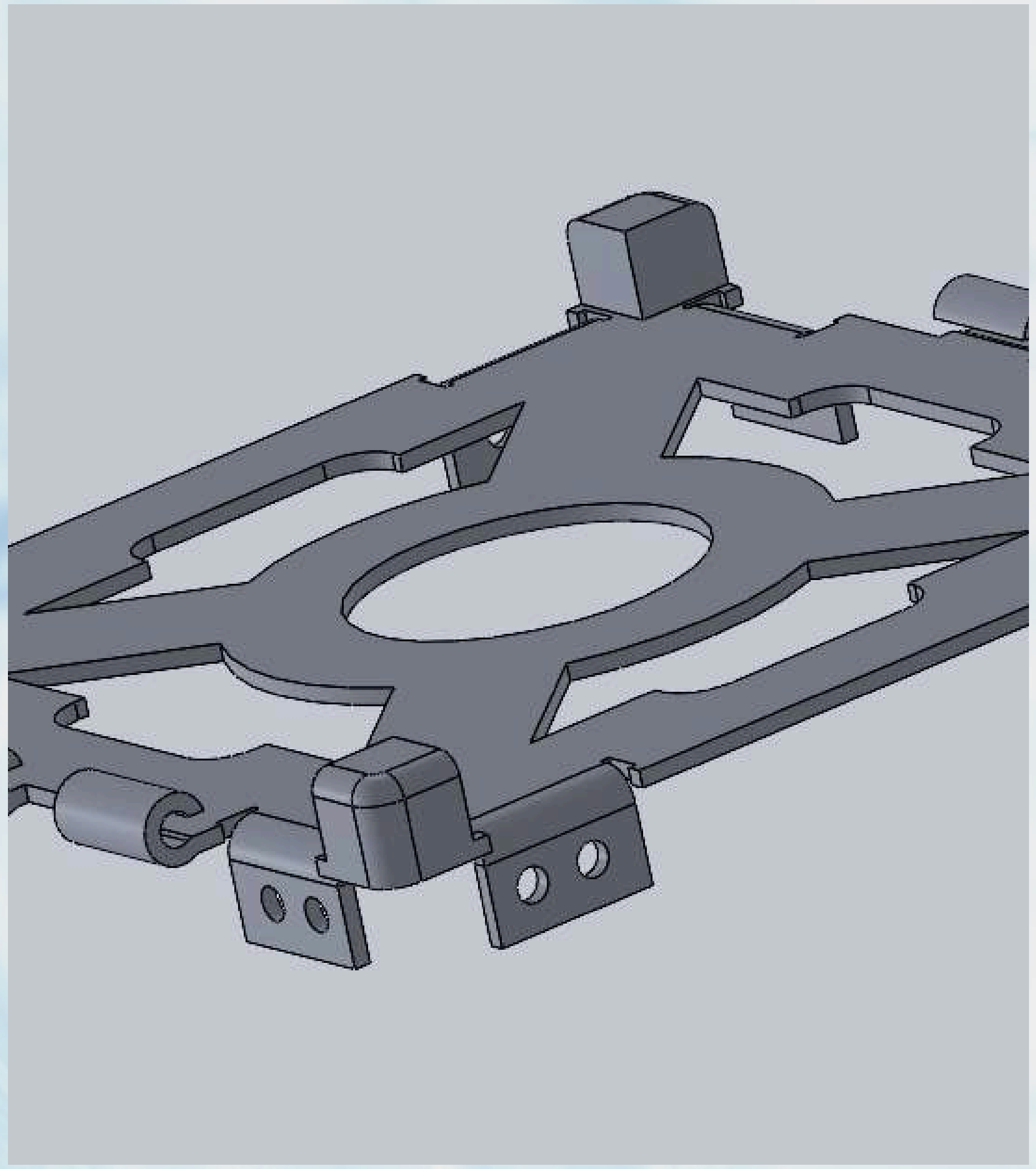
Despite the accuracy of AI systems, they are useless without a structure that protects them and provides a suitable environment to operate in space. This highlights the importance of the mechanical structure, which enables satellites to perform their missions by safely delivering these systems into orbit.

CubeSat Design and Deployable Solar Panel System

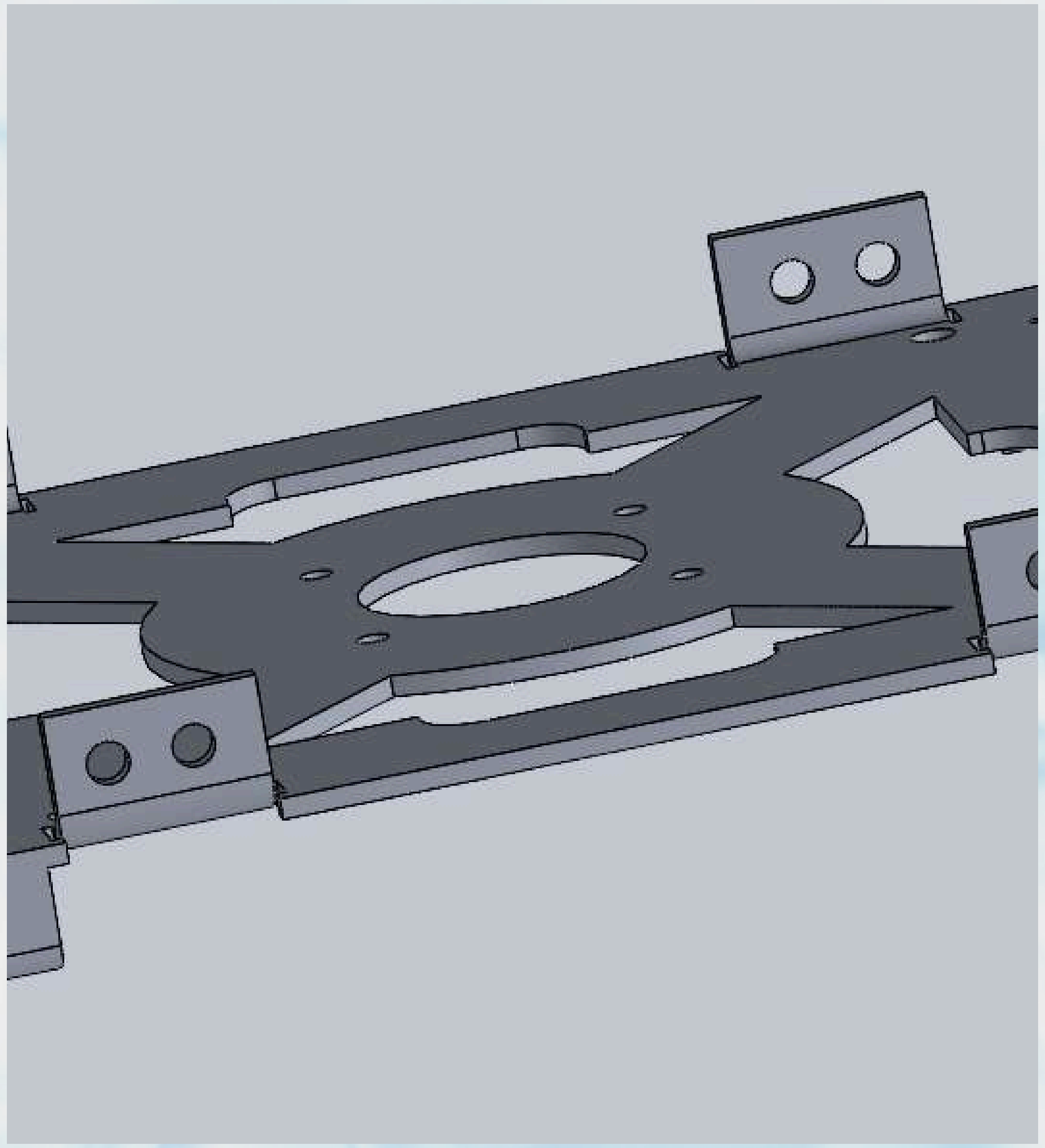
1. CubeSat Design

- A 2U CubeSat was chosen to provide the required space for various systems.

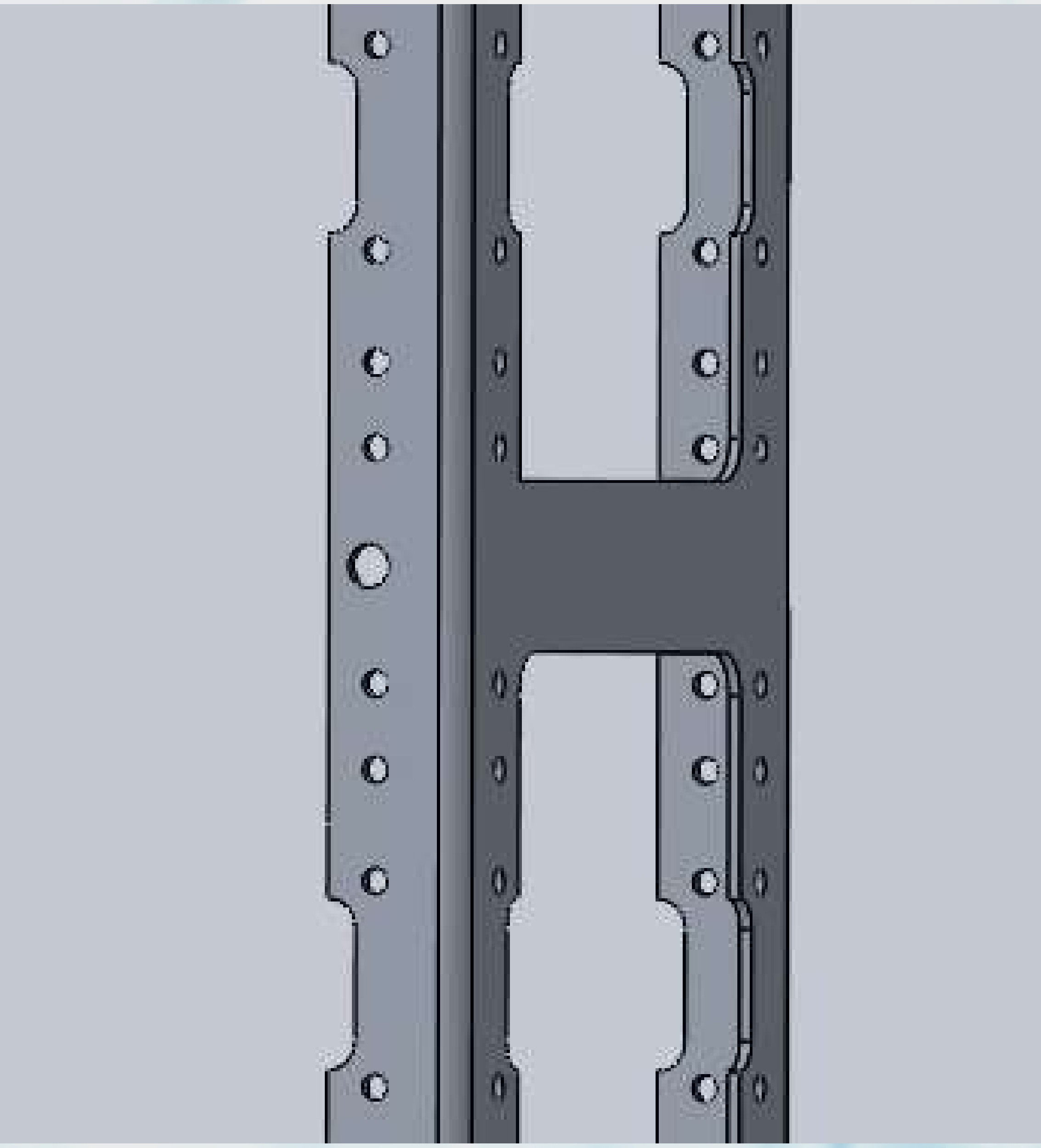
- Its structure is made of 6061 aluminum alloy due to its light weight, strength, and corrosion resistance, making it suitable for withstanding launch and orbital conditions.



Top



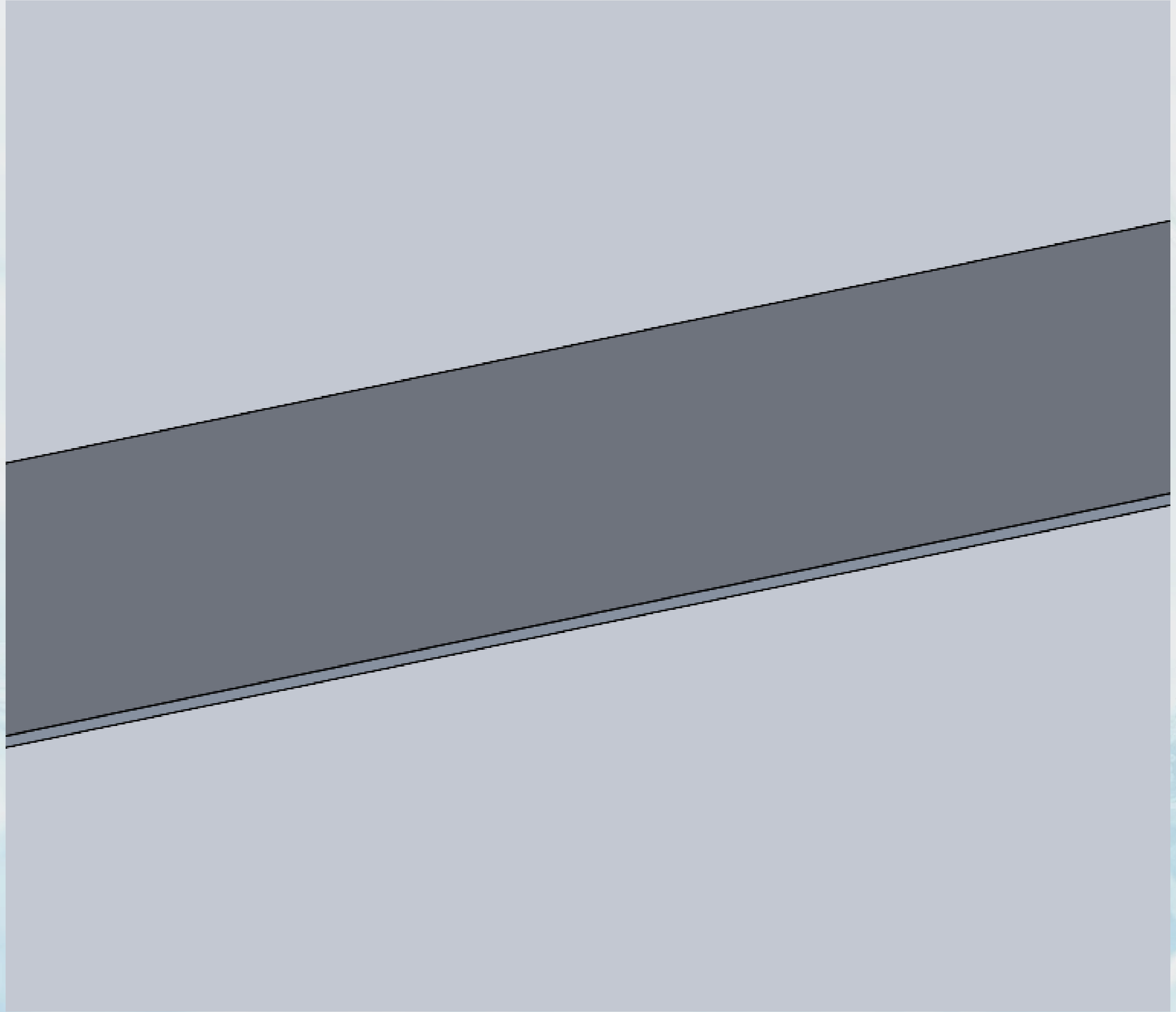
Bottom



Side

2. Deployable Solar Panel System

The CubeSat was equipped with solar cells capable of charging the batteries, which contributed to extending the operational lifetime of the mission. This allowed the satellite to rely not only on stored energy but also on the ability to recharge its batteries while in orbit.

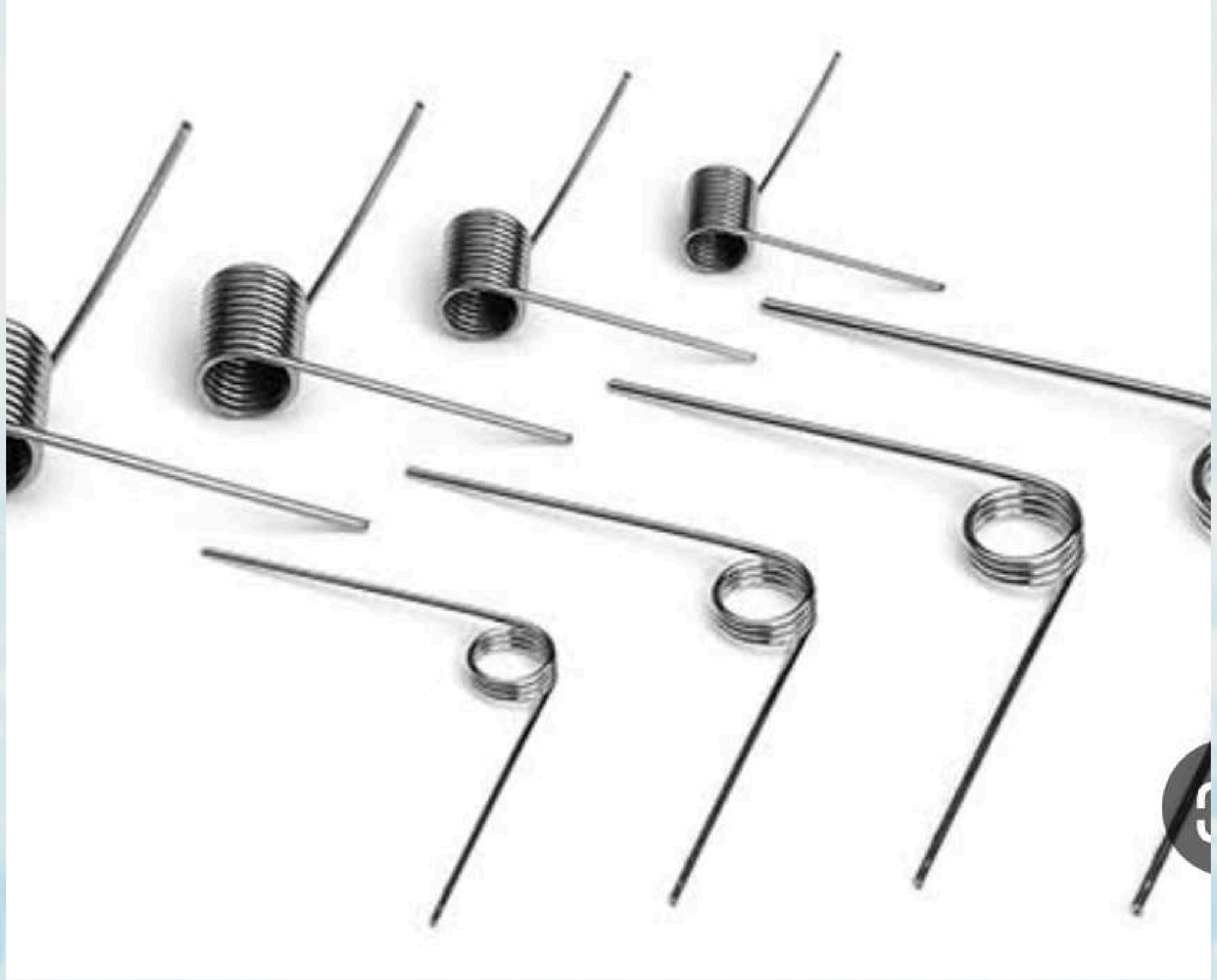


Comparison between springs and electric motors

	using a heating resistor	activate the motor
Motion Control	No precise control – panels open all at once	Precise control of angle and speed of deployment
Reusability (open/close)	Not possible – one-time deployment only	Possible in some systems that allow multiple opens/closes
Complexity	Very simple, with few moving parts	More complex – requires motor, gears, and control systems
Weight	Very lightweight	Heavier due to motor components
Power Consumption	Very low – only needed for heating the resistor	Higher – requires power to run the motor
Reliability	Very high – not dependent on electronics	Medium – depends on motor and electronics
Space Environment Tolerance	Good – not significantly affected by vacuum or radiation	Can be affected by radiation or temperature fluctuations

Torsion spring

A decision was made to use a torsion spring in the system due to its ability to store energy through twisting and to generate torque when its arms rotate around its axis.



Comparison between Spring Placement at the Center vs. at the Edges

Item	Spring in the Center	Spring at the Edges
Mechanical Force Distribution	Force is concentrated at a single point, which may lead to uneven deployment	Force is applied at the edge, providing better torque and rotation
Stability During Deployment	May cause tilting or misalignment during deployment	More stable deployment if placement is well-aligned
Ease of Design and Assembly	Requires precise central design	Often easier to install and maintain
Accuracy of Opening Angle	Requires fine-tuned torque to achieve exact angle	Provides better torque, especially with multiple springs
Cost and Number of Components	Fewer springs needed (only one)	Requires two springs (one at each edge)
Reliability	If the central spring fails, the whole mechanism may fail	Using two springs increases redundancy and reliability

Spring Design Calculations

Given:

Mass of solar panel (m) = 85 g = 0.085 kg

Gravitational acceleration (g) = 9.81 m/s^2

Length of panel (r) = 207 mm = 0.207 m

Opening angle (θ) = 90° = 1.57 rad

Material: 6061 aluminum

Young's Modulus (E) = 193×10^9 Pa

Number of active coils (N) = 5 • Wire diameter (d) = 5 mm = 0.005 m

1. Force due to Gravity: $F = m \times g = 0.085 \times 9.81 = 0.83385$ N

   Required Torque: $\tau = F \times r = 0.83385 \times 0.207 = 0.1726$ N·m

   Mean Coil Diameter (D): Using the torsion spring formula:

   $\tau = (E \times d^4 \times \theta) / (10.8 \times D \times N)$ Rearranging to find D:

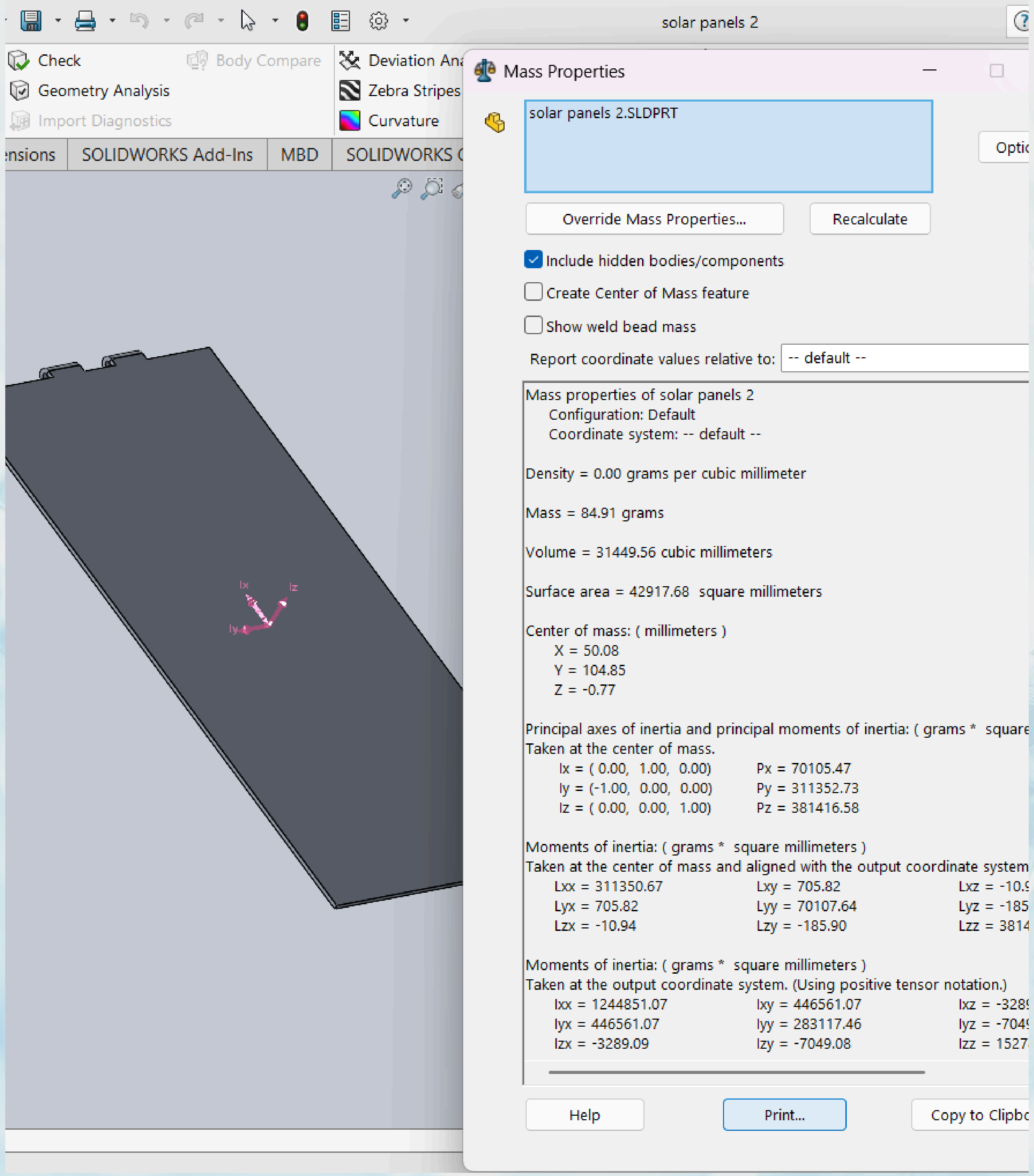
   $D = (E \times d^4 \times \theta) / (10.8 \times \tau \times N)$

   $D = (193\times10^9 \times (0.005)^4 \times 1.57) / (10.8 \times 0.1726 \times 5)$

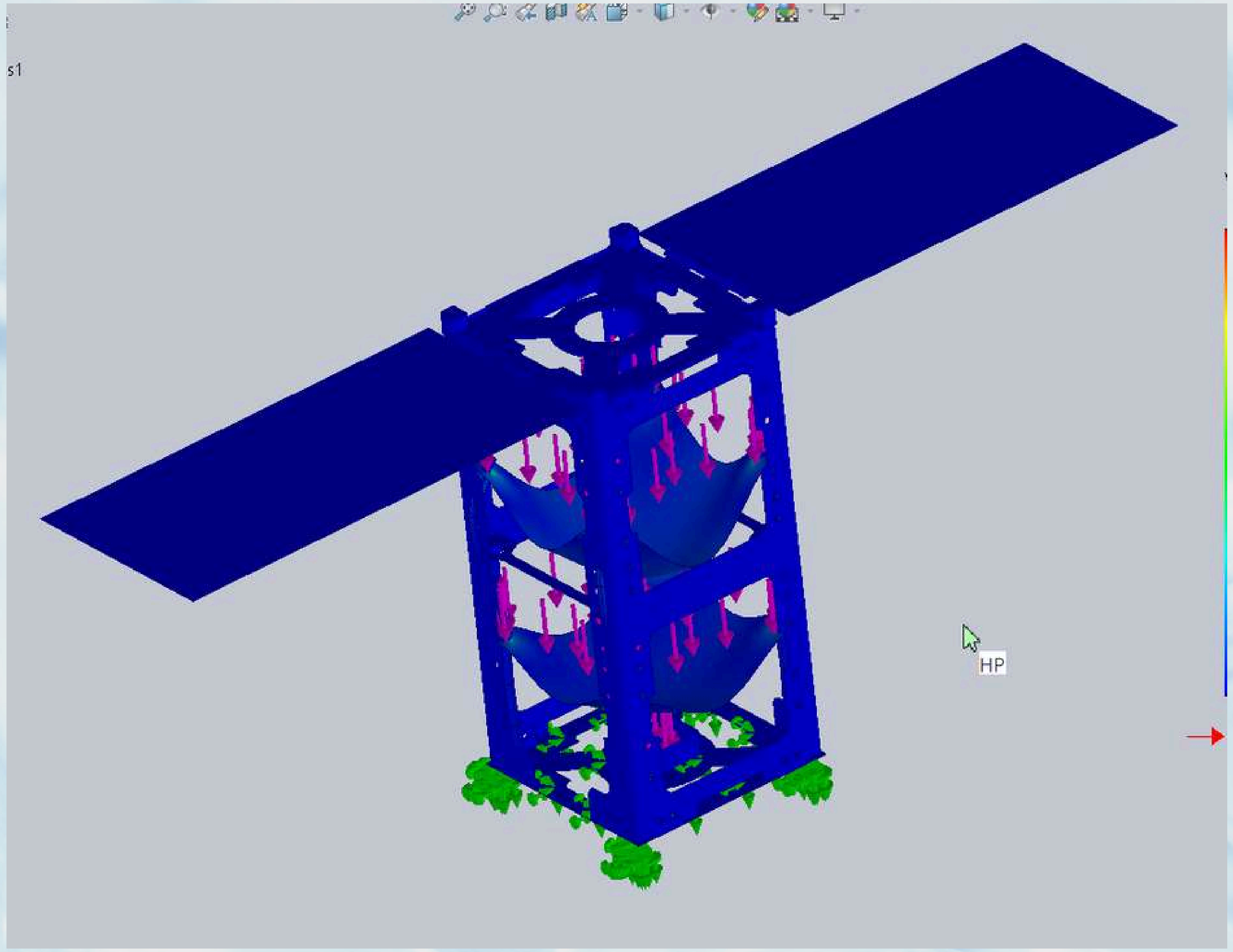
   $D = 203.5$ mm

   4. Inner Coil Diameter:

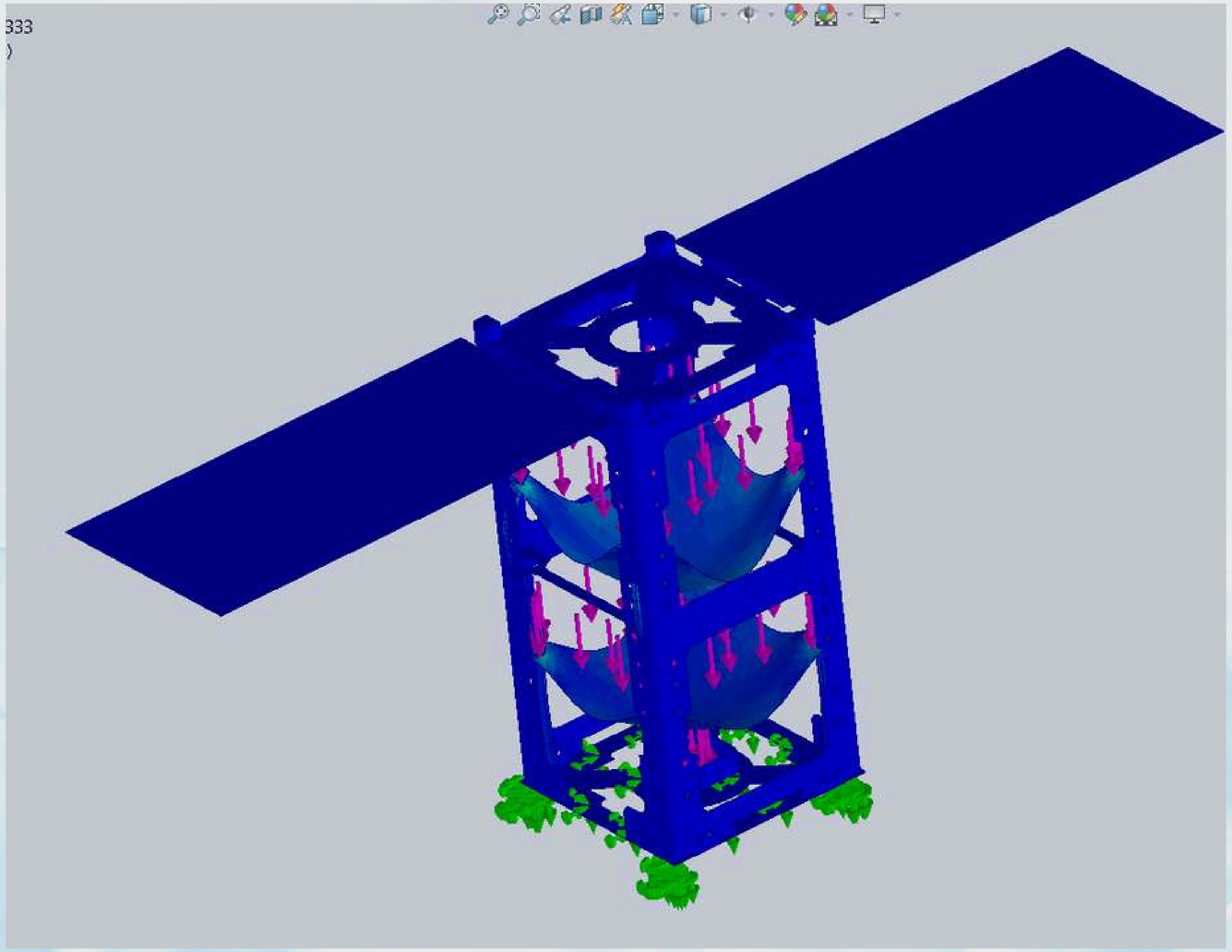
   $D\_inner = D - d = 203.5 - 5 = 198.5$ mm



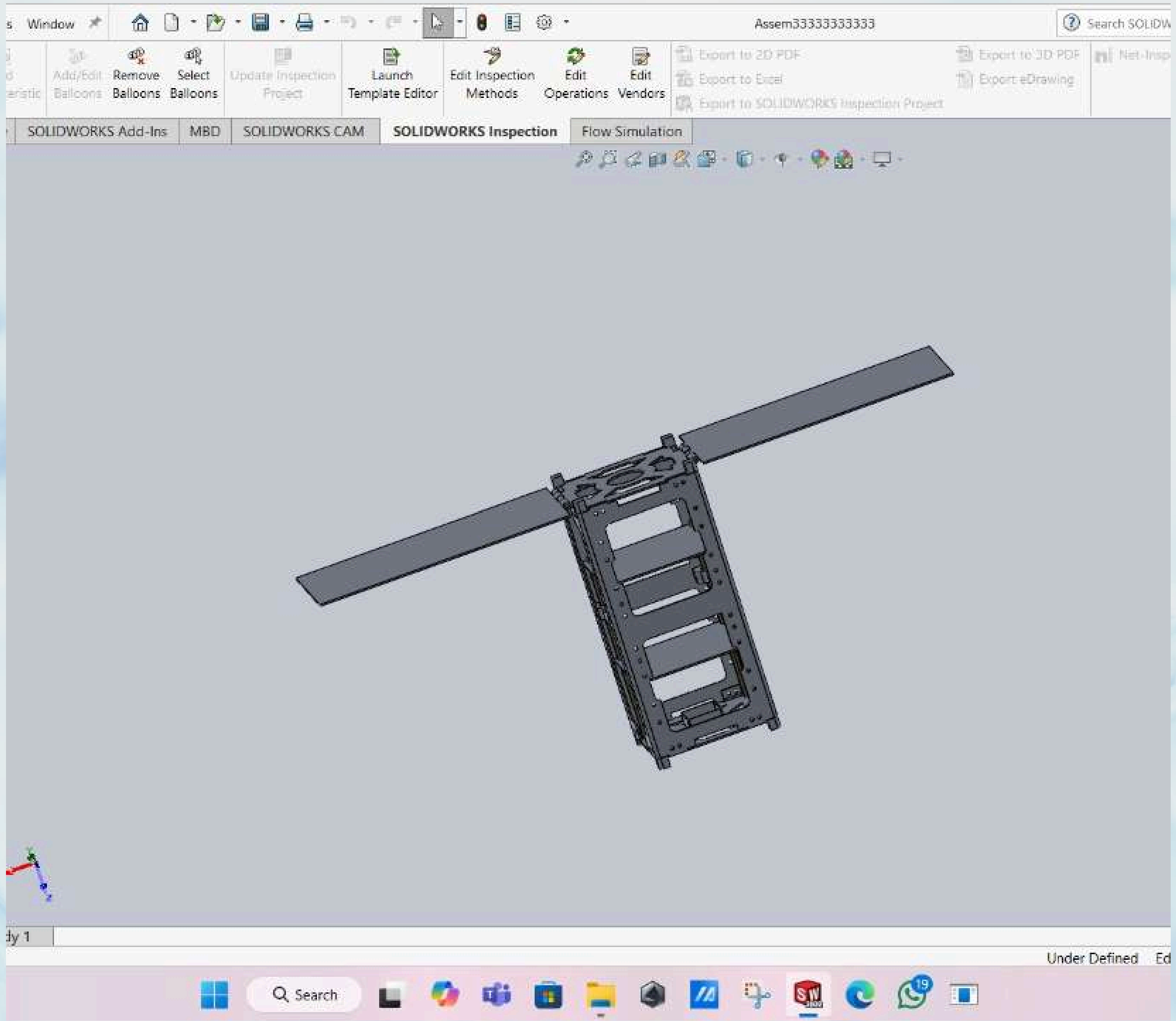
Simulation analysis in solidworks



Static stress analysis



Static strain analysis



Design in Solidworks

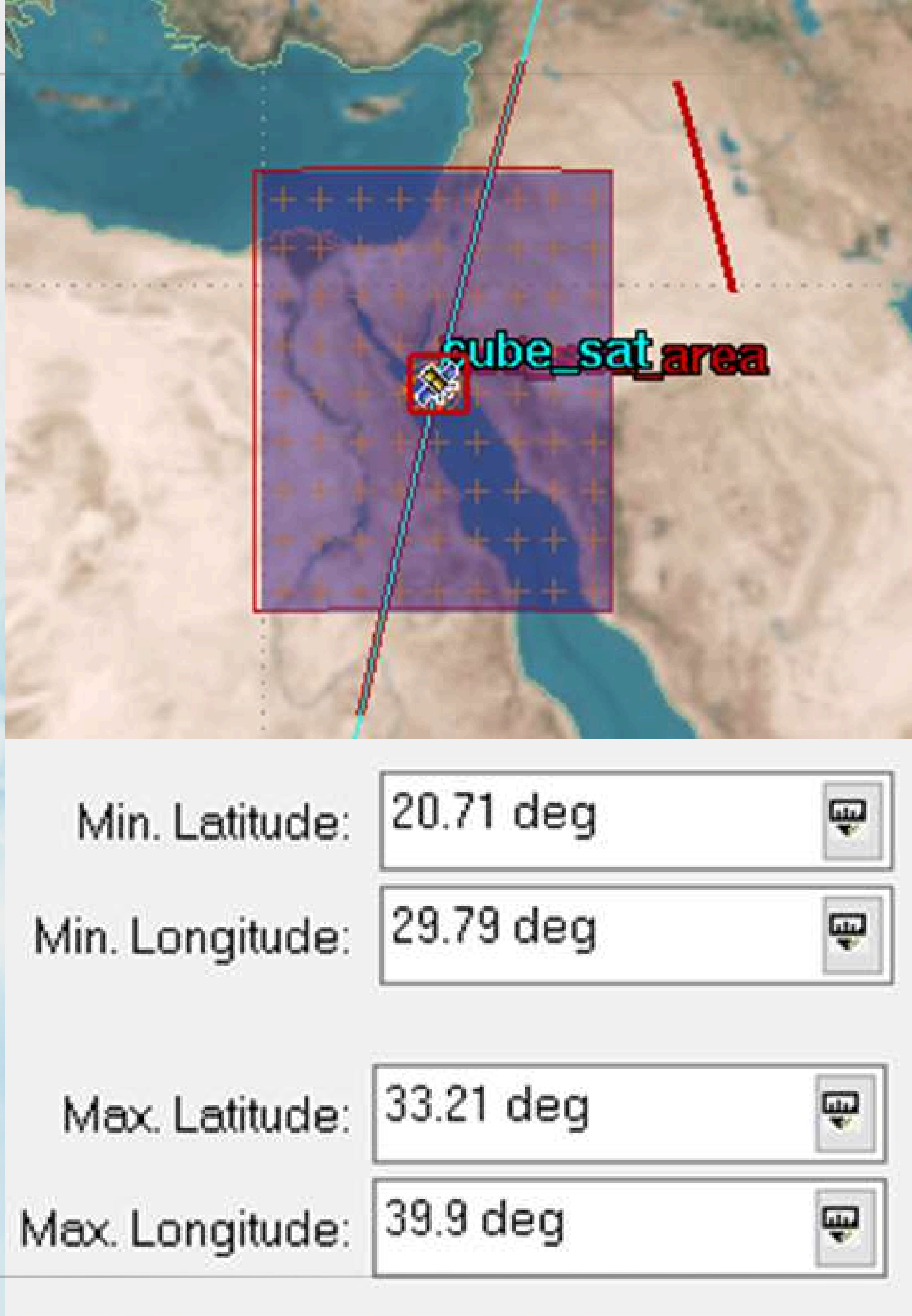
Implemented design

Simulation and Coverage Analysis with STK

Target Area Definition

Area defined as a rectangular in STK. Covers the Red Sea area region of Egypt and the northern coast of Sinai.

Used to compute access and coverage.



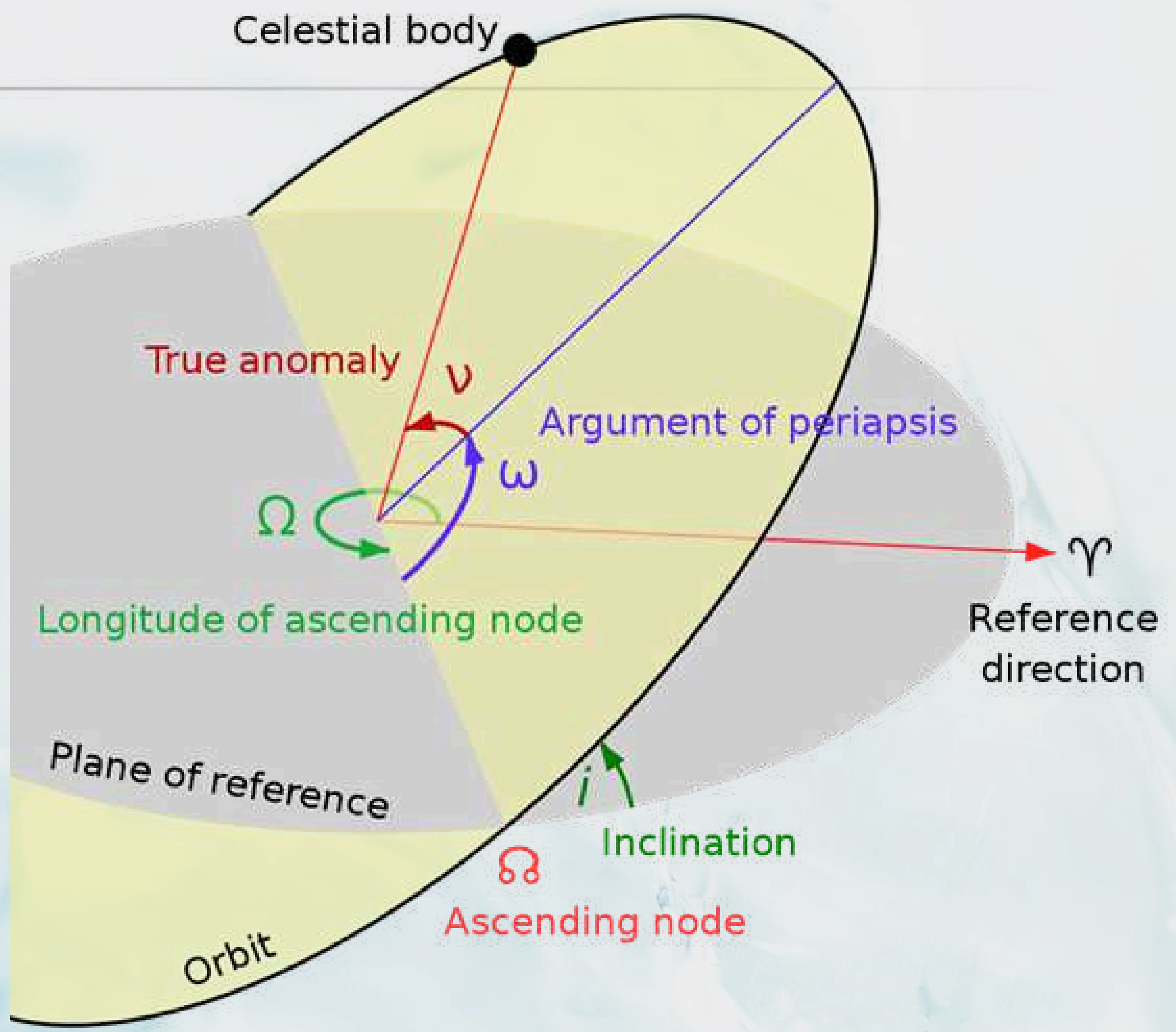
Orbit Determination

Altitude: 650 km

Orbit Type: Sun-Synchronous

LTAN: 10:30 AM

Inclination: ~97.9908°



Semimajor Axis	7028.14 km
Eccentricity	5.64944e-16
Inclination	97.9908 deg
Argument of Perigee	0 deg
RAAN	255.223 deg
True Anomaly	0 deg

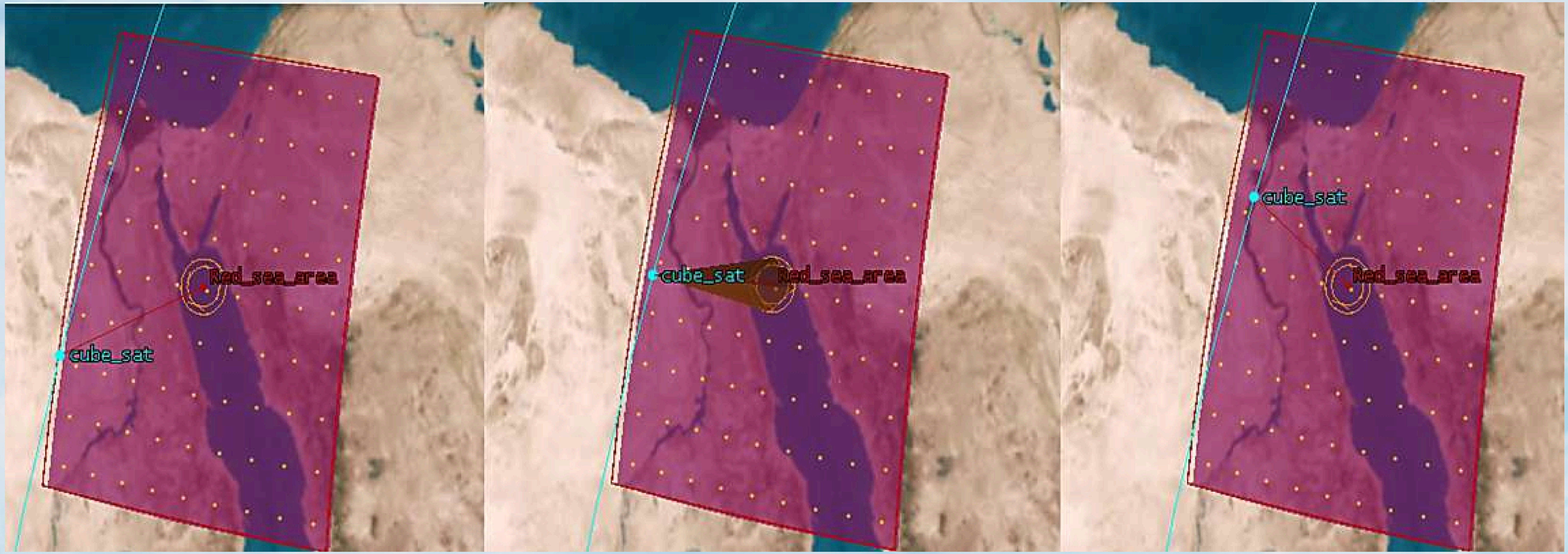
Coverage Results

Max Coverage: 100 %

Mean Coverage:3.84%

Passes per day: ~3

Revisit time: ~ 7-8 hrs



Conclusion

Orbit design is suitable for the mission objectives (Red Sea area region of Egypt and the northern coast of Sinai).

STK validated satellite visibility and data collection feasibility.

Results support integration with AI fish detection systems.

Supporting Egypt’s Blue Economy

- Largest aquaculture producer in Africa.
- AI predicts fish-rich zones (SST & Chl-a).
- Cut fishing costs by 10–15%.
- +15–25% production growth by 2028.
- Reduce imports & support local trade

Commercial, Scientific & Tech Impact

- Smarter fishing & farm management.
- Climate & biodiversity monitoring.
- Detects oxygen loss & algal blooms.
- Low-cost AI on CubeSats.
- Scalable environmental tech

Conclusion

What We Achieved?

Combined satellite data + AI to monitor SST & chlorophyll

Used LSTM & XGBoost for accurate, classified fish predictions

Built interactive maps & smart API🗺️

Tested on Raspberry Pi + simulated CubeSat system

What’s Next?

Deploy model on a real satellite

Send live data to fishermen via mobile

Expand to the Mediterranean & monitor harmful algal blooms

🇪🇬 Support national digital transformation & AI in marine resource management

Tested on Raspberry Pi + simulated CubeSat system

THANK YOU