Smart Maritime Border Alert System

Vidyashree M S[1], Nikhil Revankar[2], Keerthan V[3], Prajwal L[4]
[1]Asst. Professor, AI & ML SJB Institute of Technology
[2,3,4] Department of AI & ML SJB Institute of Technology

***Abstract*—The safety of coastal fishermen has become a significant concern due to accidental border crossings into foreign Exclusive Economic Zones (EEZs). These events are commonly attributed to rough sea conditions, limited access to navigation aids, and insufficient awareness of international maritime limits. This study presents an integrated maritime alert framework that merges GPS-based positioning, heading estimation via an MPU6050 inertial sensor, and geofencing using global EEZ boundary data. A Raspberry Pi functions as the core processing unit, analyzing real-time positional and orientation data to estimate the nearest maritime boundary using the Haversine distance model. Additionally, a heading-sensitive detection approach filters borders to consider only those aligned with the vessel's forward direction of travel. The system implements a three-level safety classification mechanism alongside an audible buzzer alert to warn fishermen as they near sensitive boundary regions. A dynamic OpenStreetMap based interface visually represents the vessel's location, orientation, and associated risk level. The proposed solution is intended to deliver a cost-effective, dependable, and easily deployable mechanism to improve safety.**

***Index Terms*—GPS Tracking, Maritime geofencing, fishermen safety, Maritime borders, Raspberry Pi, Haversine Formula**

I. INTRODUCTION

Unintentional crossings of maritime boundaries continue to represent a significant socio-economic and geopolitical challenge across many coastal regions, particularly in South Asia. Fishermen using conventional wooden or fiber vessels frequently operate without advanced navigation infrastructure, increasing the risk of drifting into foreign maritime zones. Such incidents may result in vessel seizure, legal detention and substantial financial hardship. Previous research highlights that access to low-cost navigation assistance can significantly mitigate the occurrence of such events [1]. However, the current solutions including marine GPS units, satellite tracking systems, and AIS transponders are often costly or unsuitable for small-scale fishing vessels.

To address this technological limitation, the present work introduces an affordable and intelligent maritime navigation and alert system built around a Raspberry Pi platform. The pro- posed system continuously evaluates the vessel's geographic position, heading orientation, and distance relative to predefined maritime boundaries. Unlike conventional geofencing approaches, the proposed system integrates a heading-aware border detection strategy, improving contextual relevance un- der dynamic sea conditions. By combining real-time alerts, visual mapping, and predictive distance estimation, the system seeks to substantially enhance maritime safety for small-scale fishing communities.

II. RELATED WORK

Existing research in maritime technology has largely concentrated on vessel tracking, collision prevention, and mar- itime zone surveillance. integrated fishing vessel monitoring demonstrated the effectiveness of low-cost GPS solutions in supporting nearshore safety operations [2]. Low-cost inertial measurement units have also been evaluated for marine heading estimation, showing dependable orientation tracking under moderate sea-state conditions [3]. Global geospatial datasets, including EEZ boundary maps, have been widely utilized in marine research and maritime policy analysis [4]. However, limited work exists on combining GPS, IMU heading, and EEZ geofencing specifically for fishermen safety.

Machine learning-driven zone classification techniques have been explored across various domains, with findings indicating that simple threshold-based clustering can outperform complex models when the feature space is limited [5]. Inspired by these findings, we employ a lightweight classification model tailored for embedded systems. Unlike previous systems, our work integrates real-time geospatial processing, directional filtering, safety clustering, and hardware alerts into a unified architecture.

III. SYSTEM ARCHITECTURE

The overall architecture comprises sensing modules (GPS and MPU6050), a Raspberry Pi processing unit, EEZ dataset preprocessing, a geospatial analytics engine, a safety classification module, and multimodal outputs (buzzer and live map). The architecture is designed to be modular so that additional sensors or communication modules can be added in the future without major redesign.

Fig. 1 provides an overview of the end-to-end architecture. Sensor data flows into the processing layer, which computes both absolute position and direction of travel. These values are then matched against EEZ border data and supplied to the zone classifier and alerting layer.

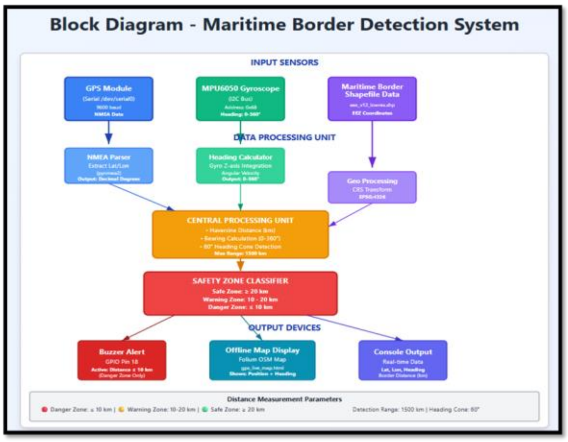

Fig. 1. System Block Diagram: Sensing, processing, geofencing, classification, and alerting modules.

IV. DYNAMIC BEHAVIOUR: SEQUENCE DIAGRAM

The dynamic behaviour of the system is governed by a continuous control loop that processes incoming sensor readings and updates outputs. As soon as the system powers on, the Raspberry Pi initializes the GPS receiver, IMU, GPIO pins, and loads the EEZ shapefile database into memory. After initialization, the system transitions into a periodic sampling loop.

During each cycle, the GPS module provides the latest NMEA sentence, which is parsed to obtain latitude and longitude. In parallel, the MPU6050 supplies gyroscope readings that are integrated into a heading estimate. The geofencing engine then evaluates distance and bearing to potential borders. The sequence of these interactions is captured in Fig. 2.

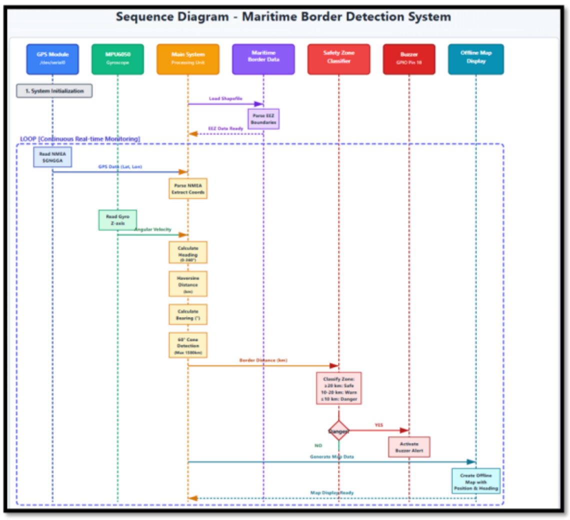

Fig. 2. Sequence Diagram: Interaction among GPS, IMU, processing engine, and alert module.

If the system classifies the situation as Safe, only visual updates are made. If the vessel enters the Warning zone, textual messages and map highlights are emphasized. In the Danger zone, the buzzer is driven continuously to demand the operator's attention. This progressive approach supports both awareness and urgency.

V. DATA FLOW MODELING

From an information-processing perspective, the proposed solution follows a structured data-flow model. Raw sensor streams are converted into structured measurements, which feed into geospatial functions and classification logic. The Level 0 Data Flow Diagram (DFD) in Fig. 3 presents this pipeline.

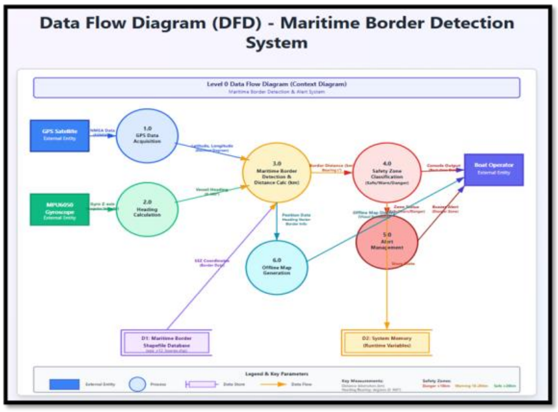

Fig. 3. Data Flow Diagram: Transformation from sensor inputs to final alerts and map visualizations.

External entities include the physical environment (satellites and sea conditions) and the fisherman as the end user. Internally, processing blocks include

GPS parsing, heading computation, EEZ distance calculation, and zone classification. Outputs are directed to both the buzzer and the map-generation module, which writes the updated HTML map file.

VI. HARDWARE INTEGRATION AND CONNECTIONS

The hardware architecture is centered around the Raspberry Pi, which provides sufficient computing resources, I/O flexibility, and compatibility with Python-based libraries. The Neo- M8N GPS module is wired through UART pins (TX and RX) for serial communication, while the MPU6050 IMU is connected via the I^2C interface (SDA, SCL). A buzzer is connected to a digital GPIO pin configured as an output.

Fig. 4 shows the complete connection diagram, including power supply considerations. A 5 V power bank can be used as the primary energy source, offering portability for deployment on small fishing vessels. Ensuring clean ground connections and proper decoupling is crucial to avoid noise-related issues with sensor readings.

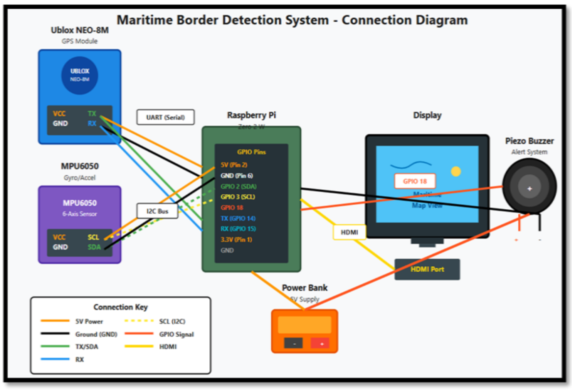

Fig. 4. Hardware Connection Diagram: Raspberry Pi with GPS module, MPU6050, buzzer, display, and power source.

The HDMI port of the Raspberry Pi connects to a small display or onboard monitor, allowing the fisherman to visualize the map in real time. This combination of physical buzzer alerts and visual guidance increases the chances that warnings will be noticed even in rough conditions.

VII. IMPLEMENTATION

The system is implemented in Python 3 on Raspberry Pi OS. Serial communication with the GPS receiver is handled using the `pyserial` library, while NMEA sentence parsing uses `pynmea2`. The IMU is interfaced via the `smbus` module. EEZ shapefile processing is carried out with GeoPandas and Shapely, enabling coordinate transformations and centroid computations.

The Haversine formula is used to compute distances be- tween the vessel's current location and the centroid of each EEZ region. Only borders within a reasonable radius are con- sidered, reducing computational overhead. A custom function then checks the angular difference between vessel heading and target border bearing to restrict consideration to a forward cone of interest.

A. Pseudocode Overview

The core execution loop can be summarized in the following pseudocode:

```
Initialize: GPS, IMU, GPIO, EEZ
dataset while True:
gps_data = read_GPS() if gps_data
is valid:
lat, lon =
parse_coordinates(gps_data) heading
= compute_heading_from_IMU() border
= find_border(lat, lon, heading)
distance = nearest_border.distance
zone = classify_zone(distance) if
zone == DANGER:
activate_buzzer() else:
deactivate_buzzer()

update_map(lat, lon,
heading,border,zone)
```

This loop is designed to be efficient enough to run continuously on a Raspberry Pi without overloading the processor or causing noticeable delays.

VIII. RESULTS AND VALIDATION

The system was validated using a combination of simulated data and controlled outdoor experiments. GPS traces approximating coastal routes were fed into the system while varying the heading angle to evaluate border detection accuracy. The Haversine-based distance measurements closely matched reference calculations, with discrepancies well within acceptable navigation tolerances for fishermen-scale operations.

The three-level zone classifier correctly flagged Warning and Danger zones as the vessel approached simulated bound- aries. In all tested scenarios, the buzzer activated within the intended threshold distance. The offline map display correctly showed the vessel's current position, projected heading vector, and textual indications of the nearest maritime border and distance remaining.

The response time from data acquisition to alert generation remained below a fraction of a second in most iterations, demonstrating that the system is capable of real-time operation in practical conditions.

IX. FUTURE WORKS

Although the current system already offers valuable safety features, several enhancements can make it more powerful and user-friendly:

- Integration of GSM or LoRaWAN to send SOS messages or position logs to coastal authorities.
- Incorporation of sensor fusion algorithms like Kalman filters for improved heading stability.
- Addition of support for offline nautical charts and bathy- metric data.
- Multi-language speech alerts tailored to regional fisher- man communities.
- Cloud-based dashboards for fisheries departments to monitor fleet positions.
- AI-driven prediction models to detect risky drift patterns before entering danger zones.

X. CONCLUSION

This work presents a practical, low-cost Smart Maritime Border Alert System aimed at protecting fishermen from accidental international boundary crossings. By combining GPS, IMU-based heading estimation, geospatial EEZ data, and real-time alerting mechanisms on a Raspberry Pi platform, the system offers an effective early warning tool that can be deployed aboard small fishing vessels. The modular design and reliance on open-source software and datasets make the solution highly extensible and customizable for different coastal regions and policy frameworks.

REFERENCES

[1] A. Sharma, "Coastal Fishermen Safety Systems," *Marine Tech Journal*, 2019.

[2] R. Menon, "Low-Cost GPS Monitoring for Small Vessels," in *IEEE OCEANS*, 2018.

[3] K. Liu, "Orientation Estimation for Marine Systems Using Low-Cost IMUs," *Sensors*, 2020.

[4] Flanders Marine Institute, "Exclusive Economic Zones Dataset," 2021.

[5] S. Gupta, "Threshold-Based Safety Clustering for Embedded Devices," *Embedded Systems Review*, 2020.

[6] J. Sinnott, "Virtues of the Haversine Formula," *Naval Navigation Journal*, 2016.