Proposed System – Working of System, Algorithm / Architecture

Working of the System:

The proposed system is a web-based application that includes a dashboard as one of its core components. where authorities or analysts can monitor vessel movements and potential oil spills in near-real time. It combines AIS vessel data and satellite imagery to detect and confirm spills using intelligent anomaly detection and image analysis.

1. User Input / Data Ingestion :

- AIS data is collected automatically via API or uploaded in CSV format (e.g., from MarineTraffic or AISHub).
- Satellite imagery is fetched on-demand from Sentinel-2 optical datasets through services like Google Earth Engine or EO Browser.

2. Preprocessing

AIS Data: Parsed and cleaned using libraries like pyAIS, pandas, and geopandas. Each vessel's movement is structured into time-series data.

Satellite Images: Preprocessed using tools like rasterio, GDAL, and OpenCV for:

Noise reduction

Radiometric correction

Land masking

3. Detection & Prediction

Vessel Anomaly Detection:

Machine learning models (e.g., Isolation Forest, LOF) analyze AIS patterns to flag unusual behavior like:

- Sudden speed drop
- Erratic turns
- Long idle time

Oil Spill Detection:

Processed satellite imagery is analyzed using:

- Thresholding (e.g., Otsu's method)
- Unsupervised clustering (K-Means, DBSCAN)
- Supervised classifiers (Random Forest, CNNs if labeled data available)

Correlation Module:

Matches AIS anomalies with satellite-detected slicks.

4. Output Display:

Results are displayed on an interactive dashboard (using Streamlit, Dash, or Leaflet) showing:

Vessel tracks and flagged anomalies

Satellite overlays with spill outlines

Metadata: vessel ID, location, time, spill extent

Alert Indicator:

- Green = No confirmed spill
- Red = Confirmed oil spill

Automatic alerts (via email/SMS/API) are sent to authorities for quick response.

5. Future Scope

Extend system to:

Predict potential spill severity or spread using weather/ocean data.

Integrate drone surveillance or real-time video feeds.

Provide recommendations for cleanup response.

Incorporate historical trend analysis for risk-prone regions.

System Architecture (Layered Overview) –

1. Frontend (User Interface / Dashboard)

- **Built with:** Plotly Dash, Streamlit, or HTML + Leaflet/JS.
- **Functions:**
  - Visualize vessel movements and flagged anomalies on a map.
  - Overlay satellite images with detected oil spills.
  - Display spill metadata (timestamp, vessel ID, location, extent).
  - Trigger alerts and provide user input (e.g., manual validation or feedback).

2. Backend (Application Logic & Coordination)

- **Built with:** Python Flask or Fast API.
- **Functions:**
  - Accepts requests for AIS analysis and satellite processing.
  - Coordinates between AIS anomaly module, satellite image analyzer, and correlation logic.
  - Handles API routes for real-time communication between frontend and backend.

3. ML/Detection Engine (Core Intelligence Layer)

- **Modules:**
  - **AIS Anomaly Detector:**
    - Uses models like Isolation Forest, LOF, or Rule-Based Thresholding.
  - **Oil Spill Classifier (Satellite):**
    - Thresholding, K-Means clustering, or supervised models (Random Forest, CNN).
- **Input Data:**

- AIS features: speed, course, timestamps.
- Satellite imagery: SAR or optical bands (preprocessed).

- **Output:**
  - Flagged anomalies
  - Confirmed oil slick regions with confidence scores

4. Data Storage & Resources (Supporting Layer)

- **Static Files:**
  - Trained ML models (ais_model.pkl, spill_model.pkl)
  - Preprocessing files (scaler, encoders)
- **External Data Sources:**
  - AIS data via MarineTraffic CSV
  - Satellite imagery from Sentinel-2 via Copernicus