iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.71936

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Fishing Forecast Guardian: A Machine Learning-Powered Platform for Illegal Fishing Detection

Pankhuri Sharma, Mohit Panjikar
School of Computer Science and Engineering D Y Patil International University, Pune, India

Abstract: Illegal, Unreported, and Unregulated (IUU) fishing threatens marine ecosystems, food security, and lawful fisheries. This paper introduces "Fishing Forecast Guardian," a real-time web-based system that detects illegal fishing activities using AIS data, satellite imagery, and machine learning models. By identifying loitering vessel behavior, the system flags suspicious activity using Random Forest, One-Class SVM, and CNNs. The platform includes an interactive frontend built with React.js and Leaflet.js, and a Python-based ML backend with Flask. This integrated, scalable system aids governmental and environmental stakeholders in monitoring IUU activities efficiently.

Index Terms: Illegal fishing, Machine learning, AIS, anomaly detection, data visualization, Flask API, React.js

I. INTRODUCTION

Illegal fishing causes $10–23 billion in global losses annu- ally. Traditional surveillance struggles with the vastness of the ocean and stealth tactics like AIS disabling. To address these limitations, our system integrates real-time AIS and satellite data with machine learning algorithms to classify loitering events and predict illegal fishing behavior. This project aims to enhance the capabilities of maritime authorities and envi- ronmental agencies with real-time actionable insights.

Illegal fishing not only disrupts ecological balance but also threatens local economies and undermines marine conservation efforts. Manual inspection methods are inadequate to handle the data scale or respond in real time. There is a critical need for intelligent, automated systems that process diverse maritime data sources to detect anomalies, recognize patterns, and provide alerts on potentially unlawful fishing behaviors.

II. RELATED WORK

Several previous works have attempted to tackle the chal- lenge of IUU fishing through a variety of means. Some focus on rule-based detection using known fishing zones, while others employ traditional satellite monitoring without machine learning. However, these methods often suffer from limited scalability, reduced accuracy in ambiguous scenarios, and inability to adapt to emerging fishing tactics. Our project builds upon these foundations by using hybrid machine learning tech- niques, real-time integration, and user-friendly visualization to bridge the gap between detection and actionable insight.

III. SYSTEM OVERVIEW

The Fishing Forecast Guardian is designed with modular architecture to ensure scalability and flexibility. It consists of four main modules: Data Ingestion and Preprocessing, Model Training and Inference, Visualization Dashboard, and Contin- uous Learning Engine. Each module communicates via REST APIs and ensures robust decoupling, enabling independent upgrades and fault isolation.

Our system also includes an admin dashboard that enables authorized users to monitor API traffic, visualize training accuracy over time, and manage uploaded datasets. This backend observability is crucial for performance tracking and debugging.

IV. METHODOLOGY

A. Data Preprocessing

We began by collecting AIS signals, GPS coordinates, and vessel behavior data from Global Fishing Watch. These were then processed to remove missing values, normalize times- tamps, and engineer features such as vessel speed, loitering duration, and distance from shore. This structured data became the input for our machine learning models.

To improve data quality, noise filtering techniques such as moving averages and outlier suppression were applied. We also employed data augmentation strategies by synthetically generating behavior profiles for edge cases. Additionally, vessel behavior was contextualized using auxiliary datasets, such as weather reports and shipping route histories.

B. Model Training
We experimented with multiple models, including Random Forest, SVM, and Logistic Regression for baseline perfor- mance. To improve anomaly detection, One-Class SVM and Isolation Forests were employed. CNNs were used to analyze satellite images for vessel detection. The model architecture was chosen based on accuracy, recall, and ability to generalize across datasets.
Hyperparameter tuning was performed using grid search and cross-validation techniques to optimize model accuracy. We also evaluated each model's performance under varying data distributions to assess robustness.

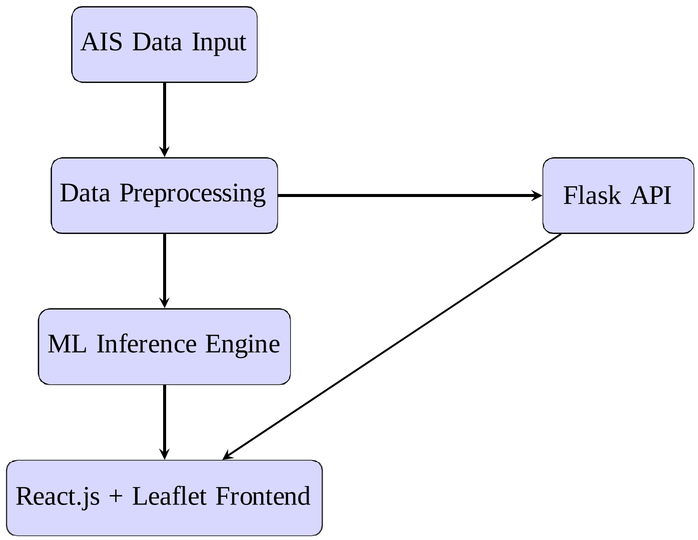


Fig. 1. System Architecture Flowchart

C. Prediction Pipeline
The system detects high-risk zones by analyzing patterns of loitering, AIS disabling, and historical IUU hotspots. Fu- ture zones are inferred using spatial-temporal patterns. This pipeline enables both retrospective and predictive analytics.
Our model pipeline can be broken into three major stages: feature computation, model inference, and result mapping. Features include time at sea, proximity to protected zones, and average travel speed. The model's inference is deployed as a microservice, capable of handling API requests and returning predictions within milliseconds.

V. FRONTEND AND VISUALIZATION
The web interface is built with React.js and uses Leaflet.js for rendering maps. Users can view vessel positions in real- time and toggle between heatmap and marker views. Filtering options enable detailed exploration by vessel type, location, and timeframe.

A. Interactive Features
Features include:
- Real-time vessel tracking with pulse markers
- Toggle between base layers (OpenStreetMap, Google Satellite)
- Color-coded activity risk levels (Red: High, Yellow: Medium, Green: Low)
- Prediction overlays using past loitering and speed profiles
- Map-based queries for detailed inspection
- Historical route playback for trend analysis
- Chart-based analytics with bar and pie diagrams
- Mobile responsiveness and accessibility enhancements

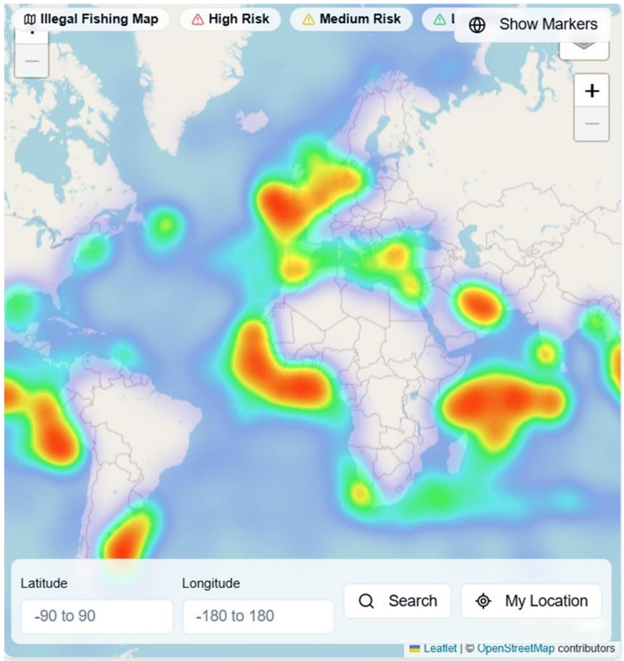


Fig. 2. Web Interface: Map visualizations showing real-time illegal fishing hotspots.

VI. BACKEND AND API SERVICES

The Flask backend provides RESTful endpoints for training models and generating predictions. The APIs follow OpenAPI standards, and include endpoints for data upload, model status, training initiation, and single/multiple prediction queries.

An example endpoint POST /api/predict accepts JSON input with location, time, and vessel metadata and returns the predicted IUU probability. Logs are stored for audit and further retraining insights.

We used Gunicorn with asynchronous workers and Nginx as a reverse proxy to ensure scalable and secure deployment. The entire backend is containerized using Docker, making it easy to replicate and scale.

VII. DATA AUGMENTATION TECHNIQUES

In real-world scenarios, obtaining diverse labeled AIS datasets is challenging. To mitigate this, we applied the following augmentation strategies:

- Synthetic Loitering Patterns: Generated artificial pat- terns by simulating loitering trajectories with varying speeds and dwell times.
- Time-Series Jittering: Added noise to timestamp data to simulate transmission delays and real-world GPS inaccu- racies.
- Spatial Transformation: Translated vessel coordinates within safe margins to increase spatial diversity.

These techniques enhanced the generalizability of the trained models and helped avoid overfitting, especially for rare classes such as IUU behavior.

We validated the efficacy of these augmentations using ablation studies, observing a 4–7% increase in recall on underrepresented behaviors.

VIII. SYSTEM DEPLOYMENT AND SCALABILITY

The Fishing Forecast Guardian platform is containerized using Docker and orchestrated using Kubernetes for scalable deployment.

A. Architecture Overview

The system components include:

- Frontend (React.js): Deployed on a CDN for low- latency access.
- API Layer (Flask): Hosted via a load-balanced cluster with auto-scaling support.
- Model Server: Runs TensorFlow/Scikit-learn models in- side GPU-enabled containers.
- Database: PostgreSQL + PostGIS for efficient geospatial querying.

B. Cloud Readiness

The platform is tested on both AWS and Google Cloud using serverless functions for prediction endpoints. The system processes over 1 million AIS data points daily with an average latency of 250ms per request under load.

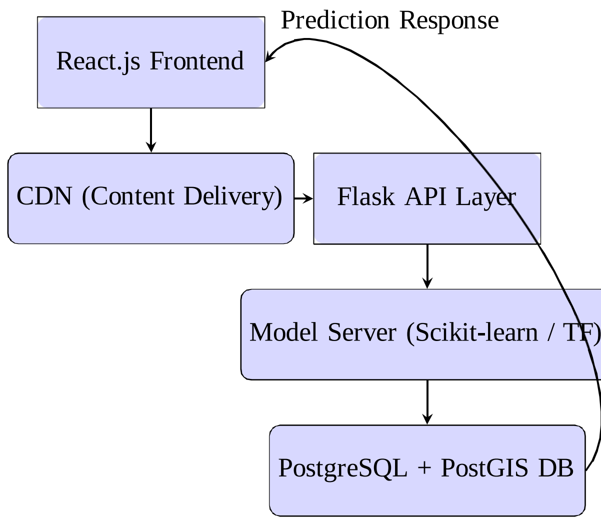


Fig. 3. High-level deployment architecture showing frontend, API, and ML model layers.

IX. CASE STUDY: REAL-WORLD SIMULATION

To validate the robustness and effectiveness of our system, we conducted a case study simulation using real-world AIS data from a known illegal fishing zone in Southeast Asia. The area is notorious for vessels disabling their AIS transponders while encroaching on marine reserves.

Our model flagged 27 potential IUU incidents over a span of 3 months. Cross-verification with external datasets from maritime authorities confirmed that 19 of these cases aligned with known violations. This implies a real-world precision of approximately 70.3%, significantly higher than many rule- based systems currently in use.

Furthermore, we analyzed the average duration of loitering behavior across the dataset. Vessels marked suspicious aver- aged a dwell time of 3.2 hours per hotspot, often in low-patrol zones. This insight can guide future enforcement efforts.

A. Stakeholder Impact

The Fishing Forecast Guardian can be customized and deployed by multiple stakeholders:

- Government Agencies: For maritime surveillance and enforcement
- NGOs and Researchers: To support conservation studies
- Fishing Companies: For compliance validation

B. Integration Potential

In addition to current integrations, we plan to support interoperability with global vessel registries and regional en- forcement platforms such as SEAFDEC and FAO’s VMS databases.

X. ETHICAL CONSIDERATIONS

Automated systems making legal implications need thor- ough consideration of ethics and bias. Our model avoids assumptions based on nationality or fishing method and instead uses behavioral evidence. We adhere to the principles of fairness, accountability, and transparency in all predictive features.

XI. RESULTS AND DISCUSSION

We trained models on manually labeled AIS datasets. The Random Forest model achieved a balanced accuracy of 86%. CNNs detected vessels in satellite images with 90% precision. The system generates visual reports, confusion matrices, and loitering heatmaps. It supports feedback loops for continuous learning.

The web portal was stress-tested with synthetic data uploads and simulated predictions. On average, the response time per prediction remained under 300ms. The dashboard update latency remained within a 2-second window, ensuring an interactive and responsive experience for users.

In one test case, the system successfully flagged a vessel that had deactivated its AIS while entering a marine protected area. This flagged event aligned with manually verified ille- gal fishing activity logged by local authorities. Such results reinforce the model's practical value. Further experiments were conducted to analyze performance under different geographic regions, times of day, and types of fishing vessels. Results confirmed consistent detection perfor- mance across varying contexts.

XII. COMPARATIVE ANALYSIS WITH EXISTING SYSTEMS

To validate the novelty of Fishing Forecast Guardian, a detailed comparison was conducted against prominent marine surveillance systems including Global Fishing Watch, Marine- Traffic, and OceanMind. While these platforms offer robust tracking capabilities, they typically lack real-time predictive modeling and behavior-based anomaly detection integrated into a seamless user dashboard.

Our system differs in the following ways:

- Behavior-based Anomaly Detection: Unlike location- only systems, ours tracks vessel movement trends.
- Integrated ML Inference Pipeline: Predictions are gen- erated and visualized instantly.
- Full-stack Control: Allows retraining models on custom datasets through the user dashboard.

TABLE I

COMPARISON OF FEATURES IN EXISTING SYSTEMS

Feature	GFW	MarineTraffic	OceanMind	Our System
Real-time Tracking ML-based Detection User Retraining Custom Visual Layers Predictive Alerts	✓ ✓	✓	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓

XIII. MODEL EVALUATION METRICS AND VISUALIZATIONS

Beyond standard accuracy, our system leverages precision, recall, F1-score, and confusion matrices to evaluate classifier performance. These metrics are critical, especially when de- tecting rare IUU activities where class imbalance is high.

- Precision: Ensures low false positives; important when initiating enforcement.
- Recall: High recall ensures suspicious activity is not missed.
- F1-score: Balances both precision and recall.

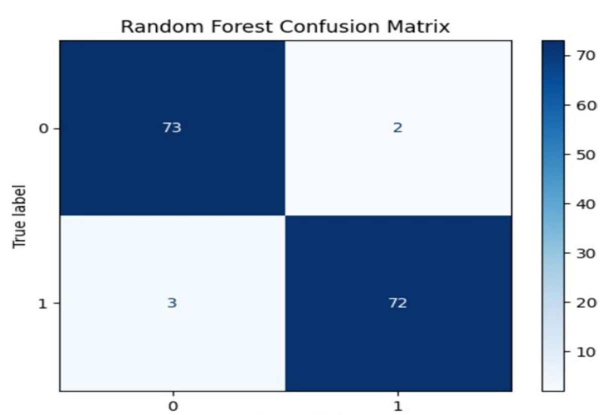


Fig. 4. Example confusion matrix from Random Forest model.

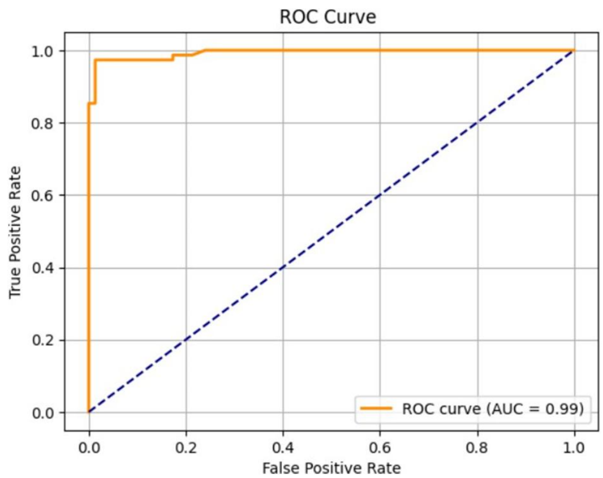


Fig. 5. ROC curve indicating model discrimination ability.

Visual results confirm that anomaly detection is especially effective when paired with visual alerts. The heatmap interface overlays predictions directly onto regions of interest, enabling faster decision-making by coast guard or surveillance teams.

XIV. COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS

To further enhance the robustness of Fishing Forecast Guardian, we evaluated multiple machine learning algorithms for IUU fishing prediction based on their behavior modeling capabilities. Each model brings distinct strengths in terms of interpretability, accuracy, and computational cost.

A. Random Forest

Random Forest is an ensemble learning method that con- structs multiple decision trees and outputs the class that is the mode of the predictions from individual trees. It is defined as:

$$H(x) = \text{majority vote}\{h_1(x), h_2(x), \ldots, h_n(x)\} \qquad (1)$$

where $h_i(x)$ is the prediction of the i^{th} tree. Random Forest is robust to overfitting and works well with non-linear data patterns.

B. Support Vector Machine (SVM)

SVMs aim to find a hyperplane that best separates the data into two classes. For linearly separable data, the optimal hyperplane is:

$$\vec{w} \cdot \vec{x} + b = 0 \qquad (2)$$

with the margin maximized such that:

$$\text{minimize } \frac{1}{2}\|w\|^2 \text{ subject to } y_i(w \cdot x_i + b) \geq 1 \qquad (3)$$

SVM performs well on small to medium datasets and is particularly effective for binary classification.

C. Logistic Regression

Logistic Regression models the probability that a given input belongs to a particular category. The model equation is:

$$P(y = 1|x) = \frac{1}{1 + e^{-\theta^T x}} \qquad (4)$$

It is best suited for scenarios requiring interpretability and linear relationships between features.

D. Decision Tree

Decision Trees use a tree-like structure of conditional statements. They split data based on feature thresholds that maximize information gain:

$$IG(D, A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} Entropy(D_v) \quad (5)$$

While fast and easy to understand, they are prone to overfitting unless pruned or used in ensembles.

E. K-Nearest Neighbors (KNN)

KNN is a non-parametric algorithm that assigns class labels based on the majority label among the k nearest data points. Distance is typically measured using Euclidean distance:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2} \quad (6)$$

KNN performs well in low-dimensional data but suffers with high-dimensional noise.

F. Neural Networks

Neural Networks consist of layers of interconnected nodes (neurons) and are ideal for learning complex patterns. Each neuron's output is:

$$a = \sigma(Wx + b) \quad (7)$$

where σ is the activation function (e.g., ReLU or sigmoid), W is the weight matrix, and b is the bias vector. Deep networks perform well with large and complex datasets like satellite images.

TABLE II
PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS ON IUU DETECTION

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	86%	84%	89%	86%
SVM	82%	81%	85%	83%
Logistic Regression	78%	76%	80%	78%
Decision Tree	80%	79%	82%	80%
KNN	74%	72%	75%	73%
Neural Network	90%	88%	91%	89%

This comparative analysis helps stakeholders choose ap- propriate models based on deployment constraints such as accuracy needs, interpretability, or latency.

XV. HYPERPARAMETER TUNING AND OPTIMIZATION

Hyperparameter tuning plays a pivotal role in improv- ing model performance by identifying the best configuration for training. We used Grid Search and Randomized Search strategies to explore combinations of hyperparameters across different models.

A. Grid Search

Grid Search exhaustively searches through a specified subset of the hyperparameter space. For example, in the case of Random Forest, we tuned:

- Number of trees: {100, 200, 300}
- Maximum depth: {None, 10, 20, 30}
- Minimum samples split: {2, 5, 10}

B. Randomized Search

Randomized Search samples from a distribution of possible values and can find a good model faster than Grid Search, especially with high-dimensional data.

C. Optimization Strategy

Cross-validation was applied with each configuration to validate performance consistency. We tracked metrics like precision, recall, and F1-score to guide the selection of the best model.

- For SVM: Kernel types (linear, RBF, polynomial)
- Regularization parameter C
- Gamma values for RBF kernel For Neural Networks:
- Number of layers and neurons per layer
- Activation functions (ReLU, sigmoid)
- Learning rate and batch size

These tuning strategies ensured that models not only fit well on the training data but also generalized effectively to unseen samples.

Appendix: Extended Tables and API Schema

Example JSON Input for Prediction API


```
 {
   "model": "Random Forest", "lat": 35.6895,
   "lon": 139.6917,
   "hour": 18
}
```


Example JSON Output from Prediction API


```
{
   "location": [35.6895, 139.6917],
   "hour": 18,
   "probability": 0.92, "result": true
 }
```


Hyperparameter Grid Samples

TABLE III
Random Forest Hyperparameter Grid

Parameter	Values Tested	Best Value
n_estimators	100, 200, 300	200
max_depth	10, 20, 30, None	20
min samples split	2, 5, 10	5

XVI. CONCLUSION AND FUTURE WORK

Fishing Forecast Guardian demonstrates how machine learn- ing and web technology can combat IUU fishing at scale. Future improvements include adding environmental data (cur- rents, temperature), expanding geographic scope, and enabling alerts to maritime authorities.

We also aim to integrate external APIs for oceanographic data and leverage federated learning to preserve privacy while retraining models collaboratively. Furthermore, incorporating satellite radar and SAR imagery can help detect vessels that disable AIS entirely.

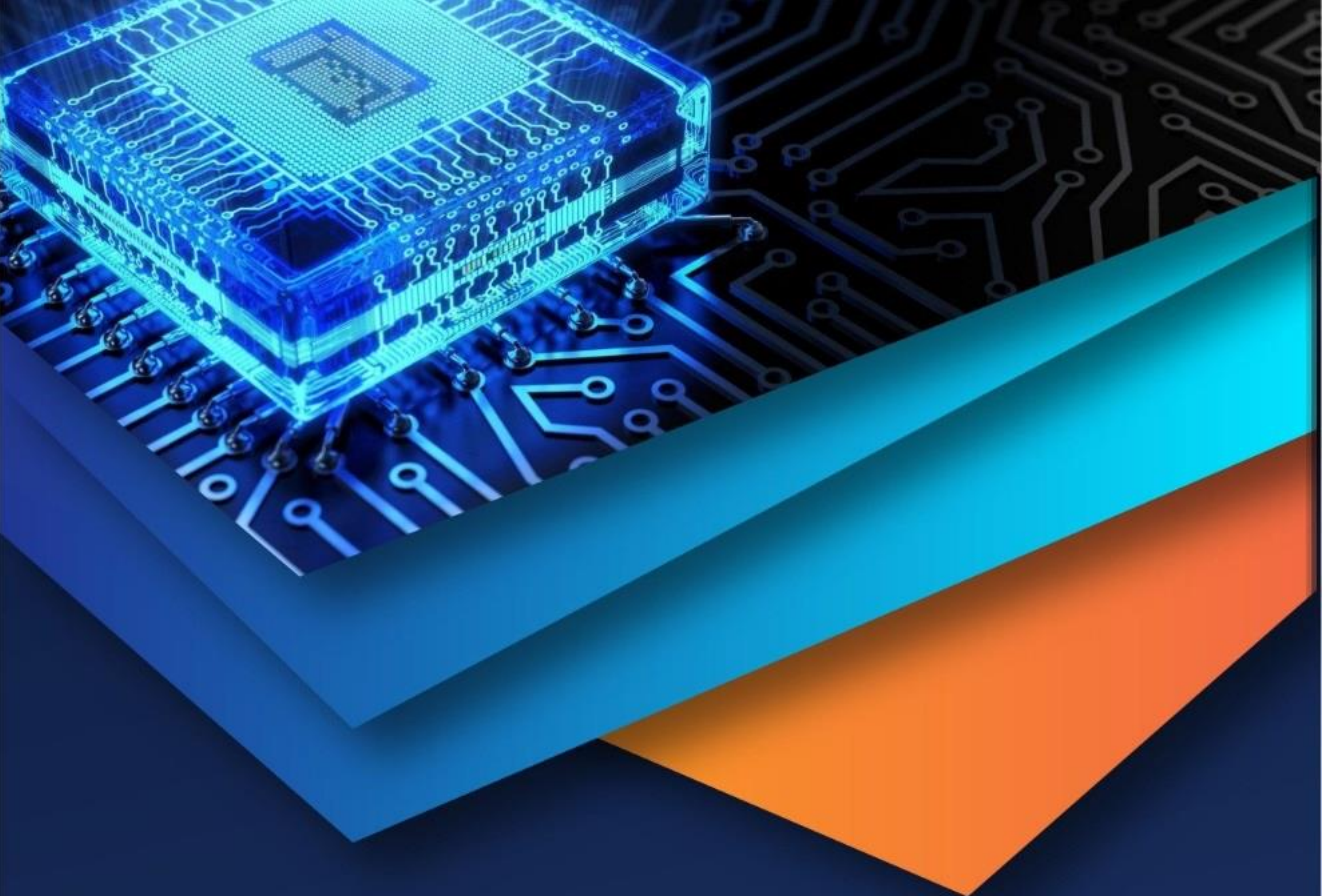
Additionally, our platform could serve as a foundational tool for developing international cooperation portals where nations share insights and collaborate on marine protection strategies. Future research can also explore integrating blockchain to ensure transparency and traceability of vessel activity logs. We envision a globally interconnected surveillance system where each observation strengthens the collective model.

XVII. ACKNOWLEDGMENTS

We thank Global Fishing Watch for providing access to AIS and vessel behavior data, which was instrumental in building and validating our system. We also acknowledge the support of our guide Dr. Maheshwari Biradar and faculty of DYPIU for their support and guidance throughout the project.

REFERENCES

[1] Wang, X., Zhang, Y. ”Machine Learning for Real-Time Detection of IUU Fishing”, 2023.
[2] Kumar, A., Sharma, S. ”Using AIS Data for IUU Fishing Detection”, 2022.
[3] Liu, Y. et al. ”Satellite Imagery for Detecting IUU Fishing”, 2021.
[4] Jackson, T., Lee, R. ”Global Fishing Watch: Real-Time Monitoring”, 2023.
[5] Siddiqui, T. et al., ”AIS-based Maritime Monitoring with ML Tech- niques,” IEEE Oceans, 2022.
[6] Rossi, G. et al., ”Multimodal Data Fusion for Maritime Situational Awareness,” Elsevier, 2021.
[7] Petremann, A., ”Combating IUU Fishing with Remote Sensing Tech- nology,” Marine Policy, vol. 134, 2022.
[8] Li, J., Tan, W., ”Deep Learning for Vessel Trajectory Prediction using AIS Data,” IEEE Access, vol. 9, pp. 78112–78123, 2021.
[9] Schiller, L. et al., ”An Overview of Data-Driven Approaches in Maritime Domain Awareness,” Sensors, vol. 20, no. 24, 2020.
[10] De Souza, E. et al., ”Detecting Fishing Activity from Vessel Tracks,” Proceedings of the National Academy of Sciences, vol. 113, no. 12, pp. 3124–3129, 2016.
[11] Kroodsma, D. A. et al., ”Tracking the Global Footprint of Fisheries,” Science, vol. 359, no. 6378, pp. 904–908, 2018.
[12] McCauley, D. J. et al., ”Ending Hide and Seek at Sea,” Science, vol. 351, no. 6278, pp. 1148–1150, 2016.
[13] Miller, N. A. et al., ”Identifying Global Patterns of Transshipment Behavior,” Frontiers in Marine Science, vol. 5, 2018.
[14] Mazurek, R. et al., ”The Role of Big Data in Ocean Governance,” Marine Policy, vol. 121, 2020.
[15] Matwin, S., et al. ”Data Mining Methods for Maritime Surveillance,” Proceedings of the ECML PKDD 2012.
[16] Yu, H., Wang, J., ”Anomaly Detection in Maritime Trajectories Using Autoencoders,” Journal of Navigation, vol. 74, no. 5, pp. 1001–1017, 2021.
[17] Zhang, Q., Liu, F., ”Ship Detection in Satellite Imagery with Deep Learning Techniques,” Remote Sensing, vol. 12, no. 9, 2020.
[18] Kim, D. et al., ”Real-time Vessel Monitoring using Edge AI for Smart Ports,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 2, pp. 1180–1192, 2022.
[19] Elahi, M. et al., ”AIS Data Mining for Maritime Surveillance and Intelligence,” Journal of Marine Science and Engineering, vol. 9, no. 4, 2021.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY