

Detecting Anomalous Ship Movements in Indonesian Seas Using Convolutional Neural Networks

Fikri Baharuddin^{1*}, *Daniel Hary Prasetyo*¹, and *Vincentius Riandaru Prasetyo*¹

¹Informatics Engineering, Faculty of Engineering, University of Surabaya, Indonesia

Abstract. Detecting unusual ship movements is a crucial feature of maritime surveillance, particularly in Indonesian waters, where illegal fishing, unauthorized resource exploitation, drifting ships, and unauthorized navigation pose significant threats to safety and security. This research proposes a Convolutional Neural Network (CNN)-based methodology for categorizing ship movement behaviors into two classifications: drifting and non-drifting. The dataset has 79,200 image-based samples, uniformly divided between the two categories. The proposed model is trained and tested using accuracy, recall, precision, F-score performance metrics. The experiment shows that the resulting model successfully classifies the movement of the ship well. This is evidenced by a testing accuracy of 0.98, a precision of 99%, a recall of 95%, an F-score of 97%, indicating that the CNN was highly accurate and robust, suggesting it could be utilized in real-time maritime anomaly detection systems.

1 Introduction

The Indonesian archipelago is one of the busiest marine locations in the world, and ships can move in ways that are against the law or strange. Maritime transportation is particularly vital to the global economy, as nearly 90% of world trade occurs by sea[1]. The thousands of islands and vital maritime routes, such as the Malacca Strait, make Indonesia's waterways both strategically essential and challenging to navigate. There are always flaws due to the high level of marine activity. This is especially true when ships move in unexpected ways that could put people in danger, allow illegal fishing, smuggling, or even piracy. Finding unusual items, such as ships that have been left behind or are adrift, is crucial for more than just safety at sea. It also helps keep the law, stop disasters, and keep the country safe. However, it's not easy to locate unusual things in a vast, ever-evolving world from a technological perspective.

In the past, maritime anomaly detection relied on manually setting criteria for factors such as a ship's speed, direction, and position. The reason for this is that standard rule-based procedures have been applied. These systems are easy to use; however, they don't always consider how dynamic and nonlinear the marine traffic is[2], [3]. When weather or sea conditions change, rule-based methods tend to perform poorly. They also don't work very

* Corresponding author: fikribaharuddin@staff.ubaya.ac.id

well with datasets from multiple sensors that are not the same. As a result, research has shifted toward data-driven methods, particularly machine learning (ML) and deep learning, which can more effectively model complex patterns.

Recent advancements in deep learning have revolutionized the detection of anomalies in various sectors, including marine surveillance. CNNs, or convolutional neural networks, are very good at identifying images and doing other things like finding ships, forecasting their routes, and finding unusual items at sea[4], [5]. CNNs are useful for marine challenges because they can automatically find hierarchical structures in raw photos. This means they can pick up both low-level data (such as shapes and forms) and high-level semantic patterns (like movement paths). Due to this ability, prior solutions that rely on manually created features are no longer effective. These traits are generally specific to one field and may not apply to other datasets.

Numerous studies have demonstrated that CNNs and other deep learning frameworks are effective for marine surveillance. For example, deep learning has been used to detect strange ship trajectories by merging it with data from an automatic identification system (AIS)[6]. Satellite imagery and synthetic aperture radar (SAR) can also be utilized to identify, track, and categorize vessels[4], [7]. These strategies have been effective in controlled conditions, but they often rely on multimodal data sources or trajectory-based features that aren't always available in real-time situations. For example, ships that are breaking the law can intentionally cut off their AIS signals, making it challenging to collect high-resolution satellite data due to the high costs and lengthy processing times. So, using these kinds of inputs could make it harder for algorithms that look for anomalies to operate well in the real world[8].

This study aims to rectify these shortcomings by focusing on the application of image-based representations of vessel movements for anomaly detection. This research proposes a CNN-based framework trained on a substantial dataset of 79,200 samples, evenly divided between drifting and non-drifting movements. The proposed solution doesn't need trajectory characteristics or other sensor modalities because it uses visual representations of how ships travel. This makes it easier to sort things and comprehend how to do it. The binary classification setup enables a quicker computing process and facilitates its use in operational monitoring centers, where rapid and precise detection is particularly crucial.

2 Methods

This research commenced with the acquisition of data sourced from a database collecting AIS information from the Indonesian seas. The acquired data is subsequently converted into an image format based on the coordinates of a ship within a specified time frame. Upon the collection and visualization of data as an image, the image data undergoes processing using various techniques, including resizing, normalization, and image augmentation, which encompasses rotation, flipping, and transposition. Following the pre-processing phase, the processed data is segmented according to research requirements, with 80% allocated for training and 20% designated for testing. The segmented data is subsequently trained with the CNN architecture and assessed for performance through accuracy measures. The phases of this investigation are illustrated in **Fig. 1**.

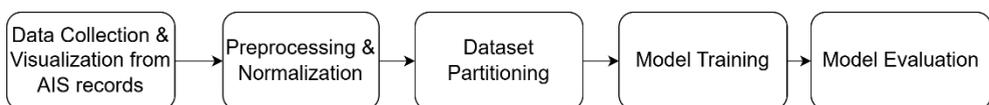


Fig. 1. Research Flow Diagram

2.1 Data Acquisition and Preprocessing

The dataset for this study was derived from Automatic Identification System (AIS) communications, which document vessel locations, speed, course, and timestamps in real-time. Each vessel trajectory was transformed into a two-dimensional picture representation, with the movement pattern over a specified time interval depicted as a grayscale or RGB image. This modification enabled us to utilize deep learning methodologies for picture classification tasks, a strategy demonstrated to be effective in marine anomaly detection scenarios[2], [9], [10]. The sample image contained in the dataset can be observed on **Fig. 2**.

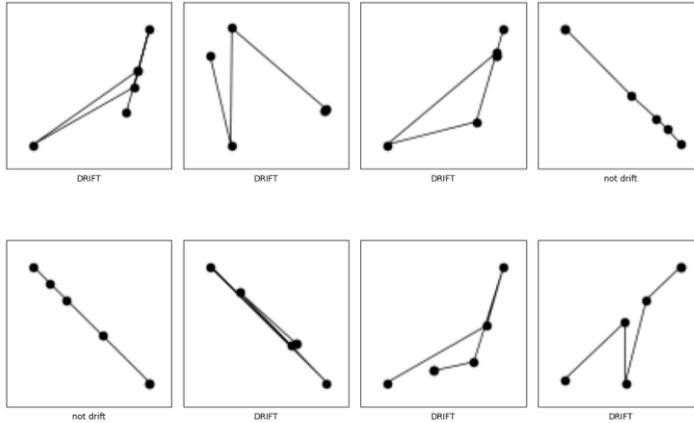


Fig. 2. Sample of Dataset

A total of 79,200 trajectory photos were amassed and categorized into two classes: drifting (characterized by anomalous movement or uncontrolled floating) and non-drifting (indicative of standard navigation). The dataset had 39,600 samples per class to mitigate bias during training. All photos were resized to $100 \times 100 \times 3$ to standardize the input dimensions and standardized to a range of $[0, 1]$. Furthermore, fundamental data augmentation methods, including random rotations, horizontal flips, and minor translations, were employed to enhance generalization and mitigate the risks of overfitting [7].

2.2 Design of CNN Architecture

A sophisticated convolutional neural network (CNN) was developed to categorize vascular movement patterns utilizing trajectory photos. As seen in Table 1 The design employs a sequential arrangement of convolutional layers, each succeeded by Batch Normalization, Max Pooling, and Dropout to enhance training stability and mitigate overfitting[11], [12]. Batch Normalization expedites convergence and alleviates internal covariate shift, whereas dropout randomly turns off neurons to mitigate overfitting [11]. The model consists of four convolutional blocks, followed by a fully connected (dense) block. The last layer employs a softmax activation function to generate probability outputs for the two categories (drifting versus non-drifting).

Table 1 The CNN architecture summary

Block	Main Layers (Summary)	Output Shape	Purpose
Input	—	(100, 100, 3)	RGB trajectory image input
Conv Block 1	Conv2D(32) + BN + MaxPool + Dropout	(49, 49, 32)	Low-level feature extraction
Conv Block 2	Conv2D(64) + BN + MaxPool + Dropout	(23, 23, 64)	Intermediate pattern detection
Conv Block 3	Conv2D(128) + BN + MaxPool + Dropout	(10, 10, 128)	High-level spatial features
Conv Block 4	Conv2D(256) + BN + MaxPool + Dropout	(4, 4, 256)	Rich spatial representation
Flatten	Flatten()	(4096,)	Feature vector generation
Dense Block	Dense(512) + BN + Dropout	(512,)	Compact learned representation
Output	Dense(2) + Softmax	(2,)	Binary classification output

This hierarchical framework enables the network to acquire more abstract representations of vessel trajectories, ranging from fundamental shapes and motion indicators in initial levels to intricate drifting patterns in deeper layers [4], [13].

The model was executed utilizing TensorFlow/Keras and trained comprehensively with the subsequent configuration listed at **Table 2**. The Adam optimizer was selected due to its flexible learning rate and effective management of sparse gradients, rendering it ideal for deep learning models[13]. A learning rate scheduler (ReduceLRonPlateau) was utilized to reduce the learning rate by half when validation accuracy plateaued, and early stopping terminated training if no enhancement was detected after 10 epochs. Combining this method can enhance convergence, mitigate overfitting, and augment model robustness[14].

Table 2 Model Configuration

Parameter	Configuration
Optimizer	Adam
Initial Learning Rate	0.001
Loss Function	Categorical Cross-Entropy
Batch Size	32
Epochs	30
Learning Rate Scheduler	ReduceLRonPlateau (halves the learning rate if validation accuracy stagnates)
Early Stopping	Stops training if no improvement is observed for 10 consecutive epochs

2.3 Evaluation Metric

In accordance with the study's parameters, accuracy, recall, precision, and F-score were designated as the principal evaluation metrics, defined as the ratio of correctly identified samples to the total number of samples.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F - Score(F1) = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (4)$$

$$False Positive Rate (FPR) = \frac{FP}{FP+TN} \quad (5)$$

$$False Negative Rate (FNR) = \frac{FN}{FN+TP} \quad (6)$$

Referring to Equation 1, TP and TN represent true positives and true negatives, respectively, while FP and FN represent false positives and false negatives, respectively. Accuracy was assessed on both the validation and test sets to determine the model's generalizability. The training and validation accuracy curves were watched to identify indications of underfitting or overfitting[15]. In addition to observing the accuracy level of the model, this study also focuses on the precision, recall, and F-Score metrics. The formula used to measure precision is presented in Equation 2. The equation used to measure recall is presented in Equation 3. As for the measurement of F-score, an observable formula is used in Equation 4. To calculate the proportion of prediction errors, this study employs the formula for measuring false positive rate and false negative rate, as shown in Equations 5 and 6.

3 Results and Discussion

The proposed CNN model was trained on a balanced dataset comprising 79,200 images, with a distribution of 39,600 images for the drifting category and an equal number of 39,600 images for the non-drifting category. **Fig. 4** illustrates the training and validation accuracy throughout the training process. During the initial epoch, the network exhibited rapid convergence, and training accuracy increased from approximately 93.5% in epoch 1 to approximately 96.2% by epoch 2. The validation accuracy exhibited greater fluctuation, commencing at approximately 94.1% by epoch 1, declining to 92.7% by epoch 2, and subsequently surging to approximately 97.9% by epoch 3. The first increase in validation performance suggests that the proposed model rapidly acquired valuable discriminative characteristics from the images. Nonetheless, by epoch 4, the validation accuracy surprisingly declined to approximately 91.7%, despite the ongoing enhancement of training accuracy. By epoch 5, the decline became pronounced, with validation accuracy falling to approximately 68.7%, while training accuracy reached approximately 97.1%. This disparity indicates a transient training instability, probably resulting from the initially elevated learning rate. The learning-rate scheduler (ReduceLROnPlateau) reduced the learning rate by half, from 0.001 to 0.0005, following epoch 5. Decreasing the learning rate in this manner is a conventional method when training seems to stagnate.

Following the reduction of the learning rate, the model's performance significantly improved. By epoch 6, the training accuracy reached approximately 97.5%, while the validation accuracy increased to approximately 98.2%. The validation accuracy consistently increased, attaining around 98.3% by epoch 8. Consequently, reducing the learning rate facilitated more precise modifications to the network weights, reinstating optimal

performance. A slight decline was observed at epoch 9, which had a validation accuracy of 94.8%, prompting a subsequent drop to 0.00025; after this, the learning process resumed smoothly. Starting from epoch 10, enhancements were significantly more incremental. At epoch 18, the training accuracy reached approximately 98.3%, while the validation accuracy peaked at approximately 98.7%. Subsequently, both curves stabilized, with training accuracy remaining approximately 98.2–98.3%, while validation accuracy fluctuated between roughly 98.5% and 98.7%. During epochs 20–28, the training accuracy fluctuated between approximately 98.2% and 98.3%, while the validation accuracy ranged from 98.5% to 98.6%, exhibiting no notable upward trend. This plateau indicates that the network had almost achieved convergence on the task, and more training epochs produced minimal improvements.

The proposed model achieved exceptional accuracy with minimal disparity between training and validation. The final training and validation accuracies were approximately comparable, indicating effective generalization. The validation accuracy surpassed the training accuracy towards the conclusion, potentially due to the regularization effects of the dropout layer implementation. Overfitting typically manifests as an increase in training accuracy while validation accuracy remains stagnant or decreases. According to the training results in **Fig. 4**, both curves reached a plateau simultaneously, indicating that the model's capacity was fully utilized on this data without significant overfitting.

The decline in validation accuracy in epochs 4–5 and 9 is classified as transient instability (optimization transients) that generally arises when the learning rate (LR) is still relatively high. In our configuration, the *ReduceLROnPlateau* scheduler lowered the LR right after plateau detection/performance degradation. It was immediately followed by a restoration of validation accuracy (epoch 6–8) to a stable level of ~98.5–98.7%. These dynamics align with the general finding that LR adjustment reduces oscillations and stabilizes tissue convergence in the presence of Batch Normalization and regularization [11], [12]. Since any decline in performance is immediately followed by recovery and is not accompanied by a widening of the train-validation gap, the phenomenon is more accurately viewed as a temporary instability due to optimization settings than an indication of systematic overfitting. If alternative phenomena such as data shift or persistent noise labels occur, the validation curve is less likely to recover after LR adjustments; It was not observed in this experiment. Following each decline, the modification of the learning rate rectified the training trajectory, resulting in sustained improvements in accuracy. By epoch 10, the two curves had effectively converged, signifying that the model had encapsulated the primary signal within the data. Beyond this juncture, further training yielded no substantial impact. In summary, both training and validation achieved saturation at approximately 98.5–98.7% accuracy, indicating that the network has attained its generalization capability on the existing dataset. No significant gap emerged, and the model did not exhibit the typical overfitting trend, characterized by an increase in training accuracy accompanied by a decline in validation accuracy. The final accuracy (98.7%) is exceptionally high by any standard. The balanced dataset indicates equivalent performance on both drifting and non-drifting classes. The proposed model accurately recognizes both classes at comparably elevated rates. Minor enhancements, such as further regularization or new data, may be feasible. Nevertheless, the results are already nearing the theoretical limit due to the intricacy of the data. The evidence demonstrates that the proposed model acquired knowledge well with negligible overfitting, generalizing proficiently to novel examples.

Fig. 3 shows the confusion matrix in the test set (Actual normal/anomaly vs Predicted). From 2000 (balanced) test samples, the model yielded TP=948, TN=995, FP=5, and FN=52. The metrics calculated are: Accuracy = 97.15%, Precision (anomaly) = 99.48%, Recall (anomaly) = 94.80%, F1 (anomaly) = 97.08%, Specificity = 99.50%, and FPR = 0.50% and FNR = 5.20%. For normal grades, Precision = 95.03%, Recall = 99.50%, and F1 = 97.22%.

In other words, the model achieves a very low prediction error rate (5 out of 1000 non-drifting data attempts) and a very high accuracy (\approx approximately 99.5%). The majority of errors originated from FN (52 cases), i.e., detection errors in drifting data, thereby reducing the drifting class recall to \sim 94.8%. The anomalous accuracy of \approx 99.5% with an FPR of 0.5% indicates that almost all alarms triggered are truly relevant, so the potential for misclassification that can cause alarm fatigue in the command center can be minimized. Operators' confidence in system performance increases. On the other hand, the presence of 52 FNs out of 1,000 drifting events (FNR 5.2%) indicates that there is still a risk of drifting misidentification, which can operationally have a costly impact. This can be caused by the presence of noise in the AIS data, which confuses the model's computation during the classification process. The high specificity and precision indicate that the model is not easily over-triggered in regular patterns and is therefore suitable for use in various marine areas in Indonesia, as well as with multiple types of vessels.

Confusion Matrix (Test)

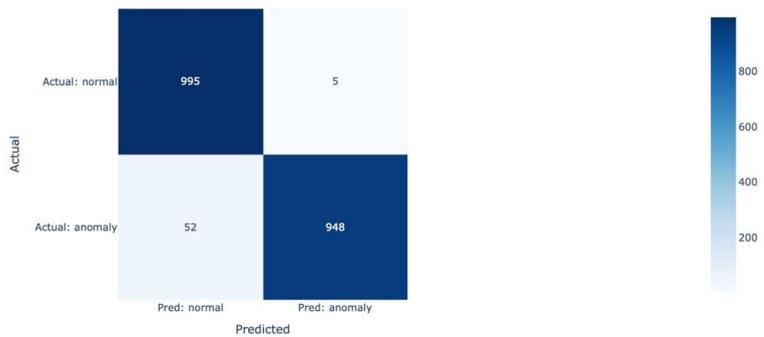


Fig. 3. Confusion Matrix of Testing Result

The proposed model's elevated classification accuracy has significant ramifications for maritime surveillance. Minor enhancements in precision can result in substantial operational benefits. For example, enhancing accuracy from 95% to 98% across numerous ship-trajectory reports could save numerous missed detections. Consequently, the proposed model's enhanced accuracy, relative to baseline approaches, directly improves the reliability of a marine anomaly detection system. Furthermore, the balanced training dataset and robust generalization indicate that the model is likely to maintain resilience across many real-world scenarios. In the Indonesian seas, characterized by diverse vessel types and itineraries, the proposed model has encountered numerous instances of both typical and anomalous behavior. Consequently, it is probable to exhibit strong performance on novel AIS data.

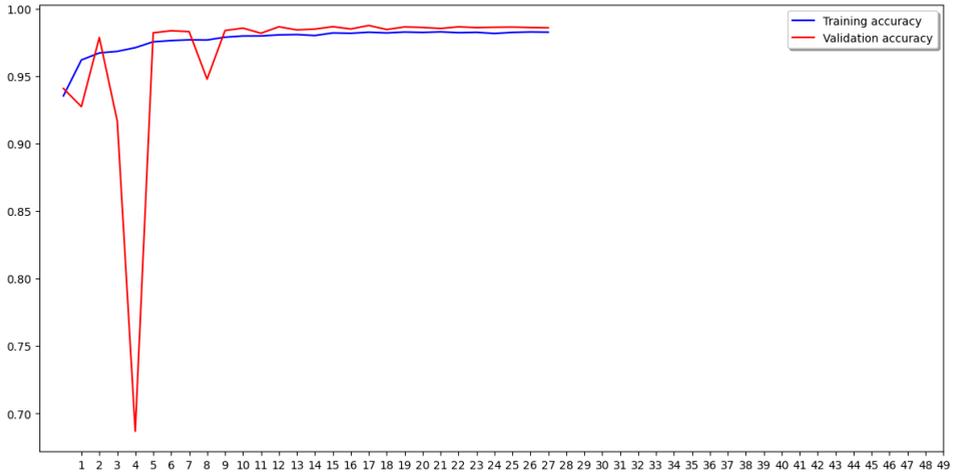


Fig. 4. Training Accuracy vs. Validation Accuracy of The Proposed CNN Model

4 Conclusion

This study introduces a Convolutional Neural Network (CNN)-based methodology for detecting abnormal ship movements in Indonesian waters, with a specific focus on identifying drifting behavior. The model was trained and tested on a balanced dataset of 79,200 trajectory photos from two classes using a well-planned experimental procedure. The findings showed that the suggested CNN worked well, with an accuracy rate of 98%. These results demonstrate that CNN architectures are effective at detecting subtle changes in ship movement and spatial patterns. From a practical point of view, maritime authorities can make the sea border safer, prevent accidents, and mitigate environmental concerns, such as oil spills or collisions, by accurately identifying drifting ships.

The CNN-based model also demonstrates how deep learning can aid in monitoring the ocean. This study, however, has many issues. There were only two kinds of information: drifting and not drifting. Marine traffic, on the other hand, exhibits a broader range of unusual behaviors, such as lingering, fishing without permission, and changing direction abruptly. Future work will focus on adding more types of anomalies to the dataset and using a stronger and more reliable algorithm to find ship movement anomalies. Future work could also investigate the use of real-time surveillance models to identify anomalous ship movements in real-time. In conclusion, the suggested CNN-based architecture is a step in the right direction for intelligent marine surveillance. It will be safer and easier to watch over ships in Indonesian waters and elsewhere.

References

1. UNCTAD, *Review of Maritime Transport 2023*. United Nations Conference on Trade and Development, 2023. [Online]. Available: <https://unctad.org/topic/transport-and-trade-logistics/review-of-maritime-transport>
2. J. Ribeiro, P. Ferreira, and J. Cunha, "AIS-based maritime anomaly traffic detection: A review," *Expert Syst Appl*, vol. 222, p. 119803, 2023, doi: 10.1016/j.eswa.2023.119803.

3. A. Stach, M. Steinmetz, and L. Sommer, "Maritime anomaly detection for vessel traffic services," *J Mar Sci Eng*, vol. 11, no. 6, p. 1174, 2023, doi: 10.3390/jmse11061174.
4. Y. Zhao, B. Xu, and W. Zhang, "Ship detection with deep learning in optical remote sensing images: A review," *Remote Sens (Basel)*, vol. 16, no. 7, p. 1145, 2024, doi: 10.3390/rs16071145.
5. A. Gupta, R. Patel, and J. Wang, "Ship detection using ensemble deep learning techniques," *Sci Rep*, vol. 14, p. 80239, 2024, doi: 10.1038/s41598-024-80239-y.
6. N. Evmides, D. Tsouros, and N. Paterakis, "Deep recurrent architectures for AIS trajectory prediction," *J Mar Sci Eng*, vol. 13, no. 8, p. 1439, 2025, doi: 10.3390/jmse13081439.
7. J. Li, Z. Liu, and Y. Wang, "Deep learning for SAR ship detection: Past, present and future," *Remote Sens (Basel)*, vol. 14, no. 11, p. 2712, 2022, doi: 10.3390/rs14112712.
8. J. Mäyrä, A. Lehikoinen, and V. Matikka, "Mapping recreational marine traffic with Sentinel-2: Constraints and opportunities," *Remote Sens Environ*, vol. 312, p. 113287, 2025, doi: 10.1016/j.rse.2025.113287.
9. M. Riveiro, G. Pallotta, and M. Vespe, "Maritime anomaly detection: A review," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 8, no. 5, p. e1266, 2018, doi: 10.1002/widm.1266.
10. K. Wolsing, H. Nguyen, and H. Frey, "Anomaly detection in maritime AIS tracks: A review of recent approaches," *J Mar Sci Eng*, vol. 10, no. 1, p. 112, 2022, doi: 10.3390/jmse10010112.
11. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, PMLR, 2015, pp. 448–456. [Online]. Available: <https://proceedings.mlr.press/v37/ioffe15.html>
12. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
13. S. Patel, R. Mishra, and S. Roy, "Deep learning-based automatic detection of ships," *Sensors*, vol. 22, no. 15, p. 5223, 2022, doi: 10.3390/s22155223.
14. X. Ren, Q. Gao, and Y. Chen, "Multi-feature fusion with CNN for ship classification," *Applied Sciences*, vol. 9, no. 20, p. 4209, 2019, doi: 10.3390/app9204209.
15. M. Syed, S. Ahmed, and A. Ali, "CNN-LSTM for vessel track prediction and anomaly detection," *J Mar Sci Eng*, vol. 11, no. 4, p. 632, 2023, doi: 10.3390/jmse11040632.