A Comparative Study of Fishing Activity Detection Approaches in Maritime Surveillance

1st Vinicius D. do Nascimento
Systems Engineering and Computer Science Program (PESC)
Federal University of Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil
0000-0002-4951-1210

2nd Tiago A. O. Alves
Mathematics and Statistics Institute
State University of Rio de Janeiro (UERJ)
Rio de Janeiro, Brazil
0000-0001-6605-6670

3rd Diego L. C. Dutra
Systems Engineering and Computer Science Program (PESC)
Federal University of Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil
0000-0003-4262-7242

4th Sandip Kundu
Electrical and Computer Engineering (ECE) Department
University of Massachusetts
Amherst, USA
0000-0001-8221-3824

***Abstract*—Maritime authorities (MA) must track fishing vessels to ensure that fishing activities are limited to permitted areas. If illegal fishing is suspected, resources must be allocated to intercept and inspect the vessels. Thus, a false flag by the MA is costly, so it is important to use accurate detection methods. We compare the accuracy and computational time of the main approaches for detecting fishing activities described in the literature, using the Global Fishing Watch (GFW) dataset. We find that Long Short-Term Memory (LSTM) neural networks achieves an optimal accuracy of 1.00, while the random forest approach comes second with an accuracy of 0.87. Given the high cost of mistakes for MA, we conclude that the LSTM's high computational cost is worthwhile.**

***Index Terms*—Illegal fishing, neural network, comparison, vessel trajectory, maritime, machine learning**

I. Introduction

Illegal, Unreported and Unregulated (IUU) fishing is one of the most serious threats to marine ecosystems because it endangers national and regional efforts to manage fisheries sustainably and protect marine biodiversity. IUU fishing takes advantage of a lack of capacity and resources for effective monitoring, control and surveillance (MCS) by developing country governments [1]. One way to detect illegal fishing is by identifying vessel behavior and the area where it is fishing.

Through Maritime Situational Awareness (MSA) it is possible to optimize the efforts between the maritime community and the maritime authorities to produce sufficient awareness to detect and address appropriate action against illegal activities in a timely manner. Using data obtained from several kinds of sensors, like Automatic Identification System (AIS), radar and Satellite Images, Machine Learning (ML) techniques can aid maritime surveillance analysts in the identification of suspicious activity. MSA can utilize these techniques and sensors to monitor the maritime domain, specially for crimes against the environment, trafficking (of arms, drugs, people, and money), and the protection of marine infrastructure.

Manuscript created March 10, 2023; revised March XX, 2023.

MSA relies heavily on sensors, and the AIS is the sensor most commonly used on ships today. Safety of Life at Sea (SOLAS) [2], a convention issued by the International Maritime Organization (IMO) and verified by the Maritime Authorities (MA), specifies that all ships over 300 gross tons are required to carry an AIS transponder. The AIS is an onboard tracking system that allows vessels to periodically report their positions. AIS's primary purpose is to prevent ship collisions. Alternatively, the MA employs this sensor to monitor its maritime domain.

In case of an eventual alert in the maritime authority monitoring system and the decision by the authorities that a vessel needs to be inspected, the alert issued by the system needs to have a high degree of accuracy. The MA needs to allocate various resources for an eventual inspection, such as specialized human resources (e.g., environmental and labor inspectors, federal revenue, anti-drug inspectors, etc.), material resources (e.g., drones, helicopters, etc.), as well as characteristics that the vessel that will be designated for inspection needs to have (maximum speed, type of radar, storage capacity, weapons, etc.). Thus, a false positive could be costly for the MA. In this study, we will investigate a model that has a high rate of true positives and permits the detection of fishing activity from the trajectory of a vessel used as input.

The detection fishing trajectories is addressed by several works using AIS sensors data in Machine Learning models. Kontopoulus [3] and Arasteh [4] utilized AIS data from vessels to reconstruct a trajectory in an image, which was then used as input in a Convolutional Neural Network(CNN) architecture. Other works, like [5] used CNN to classify vessel trajectories. In the same way, Ferreira [6] used AIS raw data to create a semi-supervised method to label the vessel trajectories and a Recurrent Neural Network (RNN) architecture to classify the trajectories in fishing and sailing. Other works, like [7] used a RNN as a classifier of vessel trajectories as fishing and non-fishing. For the vessel trajectory classification problem, the main architecture used in the RNN-based literature was the long short-term memory (LSTM). Therefore, in this paper, we will always refer to the LSTM architecture as an RNN approach. Unlike previous works that used raw AIS data directly, Pedroche [8] performed preprocessing data, transforming AIS data into trajectories, and then using the trajectories-based data as input in the models. The same approach was used in [9], using the trajectory-based data in a logistic regression (LR) to classify vessel trajectories.

However, we have not found in the literature a work comparing these approaches regarding accuracy and timing performance using the same data set.

Hence, this work propose implement two approaches to classify vessel trajectories in fishing or sailing: first, using a CNN and LSTM architectures, and a AIS raw data as input. Second, using a trajectory-based data as input for a logistic regression (LR), decision tree (DT), support-vector machine (SVM), random forest (RF) and a Neural Network (NN) models.

We want to compare these approaches and models by using the same data set and measuring how well they predict test data samples and how long it takes them to do so. The Global Fishing Watching (GFW) [10] data set [11] with fishing vessels in action will be used in combination with another data set with non-fishing vessels sailing. We use this combination of datasets to have a strong differentiation of what is a fishing trajectory or not. In the GFW dataset, there are trajectories of the same vessels fishing or not. However, this differentiation was made by the status field of the AIS receiver, which is made by the crew. In some situations, the crew may have forgotten to change the status, creating some noise in the dataset. By using trajectories from other vessels, we will have a better differentiation between fishing and other vessels' trajectories. This is a labeled data set, and for this, we will use supervised learning models, as we will describe in the next sections. The data set used in this work can be found in [12]. GFW is a website launched in September 2016 by Google in partnership with Oceana [13] and SkyTruth [14] to provide the world's first global view of commercial fishing activities [15].

Having followed this short overview, the structure of this work is as follows:

- A methods section describing what vessel trajectories are, how to build them, the data used, the preprocessing, the approaches and models used in the work;
- An experiments section describing the model parameters used in the experiments, the metrics, the data set division, and the validation strategy;
- a section on the results and discussion, in which the authors comment on the main findings and explain the main factors that led to them;
- and a conclusion section highlighting the most significant work accomplishments.

II. Methods

In this section, we will address the kinds of trajectories and how to transform the AIS data set into trajectories. Furthermore, we will explain the AIS data and the pre-processing step to use in trajectory-based data and raw data approaches. Finally, we will present the machine learning models that will be compared in this work.

A. Vessel Trajectories

Every ship has a specific function; however, most of them have the purpose of transporting cargo or moving from an origin to a destination. To achieve this goal as profitably as possible, ships try to follow optimized routes in order to save fuel and also avoid routes that present a navigational risk. Cargo ship trajectories tend to be smoother, tending toward a straight line to optimize their trajectory, a example can observed in Figure 1.

Fishing vessels have specific characteristics for fishing; in situations of normal movement from one region to another, they normally behave as cargo vessels, trying to follow an optimized trajectory. However, when the ship arrives in fishing areas and starts its target activity, it needs to perform specific maneuvers. For each type of fishing activity, there are characteristic maneuvers of the vessel that can be seen in Figure 1.

A trained operator in a monitoring situation can classify images of a vessel engaged in sailing or fishing activity. The operator needs to observe, for example, whether the trajectory follows an optimal path, the depth conditions in the area, weather conditions, whether other vessels perform activities in the area that impede navigation, etc. However, when we have thousands of images to analyze in a short time, this becomes unfeasible. Hence, in this situation, a automatic classifier is the utmost importance in maritime situation awareness systems.

Ship trajectories are composed of a set of AIS messages. Each AIS message has information about the ship's position (latitude and longitude), speed (in knots), course (in degrees), about the ship (name, Maritime Mobile Service Identity (MMSI), International Maritime Organization (IMO) and IRIN) and about the trajectory (origin, destination and status). It should be noted that some information is entered manually by the crew, like origin, destination and status. Other information, like latitude, longitude, course, and speed, is obtained automatically from GPS equipment. These manual entries can lead to erroneous information, either intentionally or unintentionally. Other works can discuss this problem [16], however, in this work we will rely on this information inserted in the data set, just cleaning up impossible positions and speed situations.

B. Vessel AIS Data and Preprocessing

The AIS is collaborative equipment installed in the vessels where self report information's about position, identity and the trajectory origin and destination [17]. The AIS messages sent by vessels are received by coastal antennas installed by VTMS (Vessel Traffic Management System), maritime companies, or government authorities. Another way to receive these messages is through low-orbit satellites used by some private companies that receive the data and sell it afterwards.

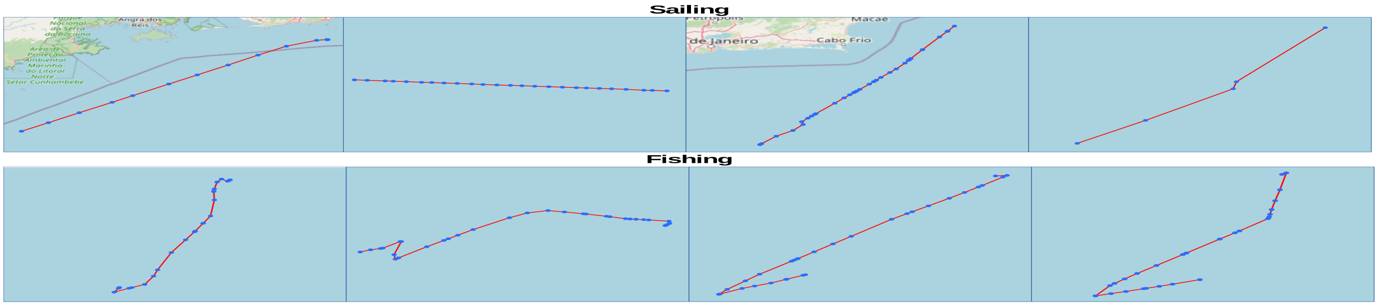

Fig. 1: Vessel trajectories constructed from historical AIS data.

This AIS message is typically used in private company or government agency MSA systems. Some AIS data sets are available to academic researchers, like in GFW [10]. The GFW is an online platform that offers a nearly real-time view of fishing activity around the globe by utilizing satellite and vessel tracking data.

In AIS data set available in GFW [10], there is AIS messages from fishing vessels in fishing trajectories. We use this GFW data set combined with other non-fishing AIS vessel messages that can be found in [12]. In figure 2, the number of AIS messages distributed by vessel type is shown.

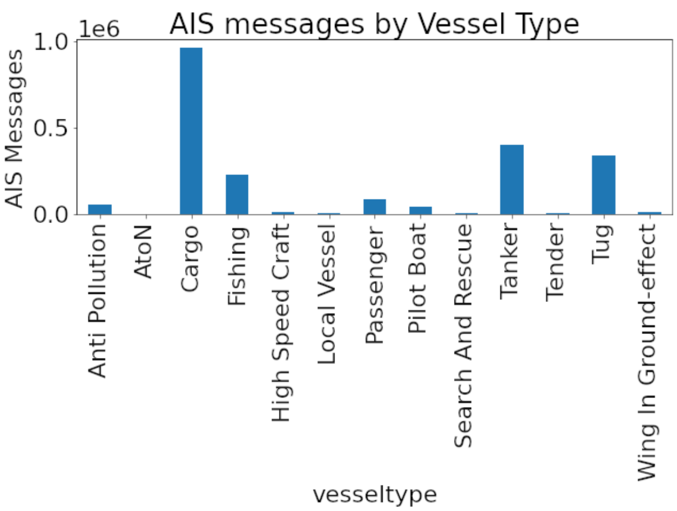

Fig. 2: AIS messages Distribution in data set

In the data set, each line contains data about location (latitude and longitude), identity (MMSI and vessel name), vessel speed and course, and a timestamp. A trajectory can be built by joining a set of AIS data from the same vessel. However, If we connect all points from a vessel in a unique trajectory, we will be considering stopping points, which is not what we want. Thus, we will consider a gap between trajectories from the same vessel, i.e., when there is a time difference between two points greater than 90 minutes, the trajectory will be split in two. This way, there will be different trajectories for the same vessel, and then it will be possible to analyze the trajectory behavior in a specific place. In the same way, if we are interested in the vessel movements, then we will only consider trajectories with speeds averaging between 1 and 50 knots and durations greater than 10 minutes.

To create these trajectories, we first created data frames in Python using Pandas [18], and then transformed the data frames into geodata frames using GeoPandas [19]. GeoPandas is a Python library used for working with geospatial data. It is built on top of Pandas and adds support for spatial data types and spatial operations. Second, we use the geo data frames to create the trajectories in Moving Pandas [20]. Moving Pandas is a Python library for handling movement data based on Pandas and Geo Pandas. Using Moving Pandas, it is possible to aggregate the AIS points in trajectories, applying metrics like speed and course average, speed and course variance, trajectory duration and so on.

This strategy was utilized in both raw-based and trajectory-based data approaches. However, only the trajectory-based approach, as described in Pedroche [8], used aggregate trajectory-related data (e.g., speed and course variance) in the models. In the raw data approach, AIS messages are grouped by trajectory, and only AIS data such as latitude, longitude, speed, course, and timestamp are utilized in the model.

Trajectory aggregated data is not obvious data that can be used in the models. In the figure 1 we can observe that fishing vessels tend to have more course variations. Hence, if we calculate the variance of course and speed of AIS points on trajectories, we can use this metric in the models. Hence, the variance can be calculated in the equation 1, where μ is the mean in all points, N the number of points and x_i the value in the i point.

$$\sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{1}$$

For example, suppose that we have two trajectories, T1 and T2, with five points each. T1 is analogous to a vessel sailing in a straight line, whereas T2 is analogous to a sinuous trajectory. The vessel course in degrees at each point in T1 is [100, 100, 100, 101, 102] and in T2 it is [90, 92, 120, 135, 150]. Calculating the course variance in both cases, we have:

$$\sigma_{T1} = \frac{1}{5}\sum_{i=1}^{N}(x_i - 100.6)^2 = 0.64$$

$$\sigma_{T2} = \frac{1}{5}\sum_{i=1}^{N}(x_i - 117.4)^2 = 555.04$$

We can perceive the sensible increase in variance value when we have a sinuous trajectory, whereas the straight trajectory has a low variance. Despite looking like a simple way to differ one trajectory from another using variance, we have other variables that can cause variance to increase. If a vessel has a long trajectory, the variance tends to increase. In an optimized path, a vessel may skirt the coast of a country, which can increase course variance. Hence, the trajectory's duration and length can interfere with variance. Another variable that should be considered is the number of points in a trajectory. Two points are the minimum for a trajectory, but not to calculate the vessel course variance. Thus, for each trajectory, we need a minimum of three points. Similarly, we can utilize the vessel speed variance.

Thus, the following data dimensions were used in the models in each approach:

- Trajectory-based data dimensions:
  - Trajectory duration: the total time of the segment.
  - Course variance: the calculated variance of the vessel's courses in trajectory.
  - Speed variance: the calculated variance of the vessel's speed in trajectory.
  - Trajectory length: the trajectory length in meters.
  - Number of points: number of points in the trajectory.
- Raw-based data dimensions:
  - latitude
  - longitude
  - speed
  - course
  - timestamp

For a fair comparison, we utilized the same trajectories in both approaches, differing only in the data used in the models depending on the chased approach. In the next sections, we will detail the models and parameter utilized in the experiments.

C. Models

To tackle this detection problem, we use two approaches found in the literature on fishing activity detection on vessel trajectories. The first uses trajectory-based data obtained after preprocessing the raw AIS data [8]. The second approach makes direct use of this raw data in the models.

The raw AIS data, such as latitude, longitude, speed, course, and a timestamp, are used to build the vessel's trajectory. Then, data about the trajectory can be gathered, such as the mean, variance, number of points on the trajectory, and length. For detecting fishing trajectories, we need classifier models that can possibly predict a class of fishing or another class of sailing. We need a supervised machine learning model because we have the trajectory data and the respective label. In the trajectory-based data approach, the following models were implemented and used in our experiments:

- Logistic Regression (LR)
- Decision Tree (DT)
- Random Forest (RT)
- Support Vector Machine (SVM)
- Neural Network (NN)

We chose these supervised models because they are widely used in the literature and we have labeled data. In the raw AIS data approach, we utilize the two models most commonly used in literature for this kind of problem:

- Convolutional Neural Network (CNN)
- Long Short-term Memory (LSTM)

The source code used in this work for these implemented approaches and models of architecture can be found in [21]. In the next section, we will give a detailed description of each model configuration utilized in the experiments.

III. Experiments

In our experiments, we utilized a data set from GFW [10] which has data about fishing vessel activity. Furthermore, we combine this data set with another, which has non-fishing vessels sailing; the full data set can be found in [12]. We clean the data set after building vessel trajectories. The average speed between 1 and 50 knots, the number of points in trajectories greater than 2, and the trajectory duration greater than 10 minutes were all filtered, and the other trajectories were dropped. After that, we got 7,336 trajectories for each class: fishing and sailing.

It is common practice to divide the data into three sets in order to estimate the generalization error of a learning algorithm: a training set, a validation set, and a test set. The training set is used to fit the learning algorithm's parameters, whereas the validation set is used to tune the model's hyperparameters. Finally, after the model has been trained and its hyperparameters have been optimized, the test set is used to estimate the generalization error of the model [22]. For our experiments, we split the data set into train, validate, and test. We utilized 80% of the data set to train and 20% to test. Inside of the 80% to train, it is was partitioned in train and validation, where 80% of this chunk was reserved to train and 20% to validation. The test data set was utilized at the end of the process to evaluate the accuracy, precision, recall, and F1 of the model.

As mentioned before, in the trajectory-based data approach, we used the LR, DT, RF, SVM, and NN models. First, we built the trajectories, and next, we grouped data by trajectories and calculated the aggregated values like speed and course average, speed and course variance, and the number of points in the trajectory. Finally, we apply the models to the trajectory data. In the raw data approach, we used the CNN and RNN models. But before apply the models, we built the trajectory points from the raw data set, which each point has the latitude, longitude, speed, and course data. We considered that a time interval of 90 minutes between two points of the same vessel is a new trajectory. In the other words, if has a gap time interval of 90 minutes, the trajectory is splitted in two.

We used a grid search for the LR, DT, RF, and SVM models, changing the parameters (such as kernel, gamma, degree, regularization, etc.) to find the best parameters for each model.

For the NN model, we created an architecture with four dense layers of sizes 32, 16, 8, and 2, respectively. After each layer, we put a 0.2 dropout to reduce the overfitting of the neural network. In the NN output, the likelihood of each class is predicted. In the first three layers, a relu activation function was used, and in the last layer, a softmax function. As an error function, a categorical cross-entropy was utilized.

In the CNN model, we utilized the same VGG16 architecture [23] as used in the Kontopoulos [3] which presented the best accuracy for classifying the fishing trajectory. First we transform the trajectories in a normalized pixel matrix of size 32x32, like in Kontopoulos's. Secondly, we put the vessel speed in RGB color array. In Kontopoulos's work the image size utilized was 224x224, but in our tests, the image with 32x32 pixels presented similar results, so we opted to use 32x32 pixels. In order to transform the trajectories into images, we first add the points into a fixed-size pixel matrix of 32x32, normalizing the X and Y of the points proportionally to the size of the dimensions. To connect the points by lines, as in Kontopoulos's, we use Bresenham's [24] interpolation algorithm. Bresenham's line algorithm is a line-drawing algorithm that produces points that approximate a straight line between two points in an N-dimensional raster. Exactly because we used this interpolation technique, we were able to reduce the dimensions of the images from 224x224 to 32x32 while maintaining similar accuracy, since the lines between the points would be interpolated in both image dimensions. Like in the NN model, the CNN model output is the likelihood of each class being predicted. An example of a sailing and fishing trajectory can be seen in the figures 3 and 4 respectively. The wider points represent the collected points of the AIS messages, while the narrower points were interpolated, completing the trajectory. In the wider points, the speed value is in the RGB field.

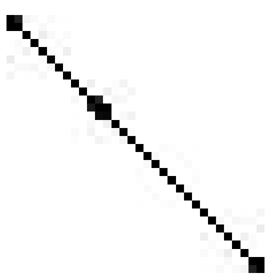

Fig. 3: AIS points of vessel in a common sailing trajectory converted to a 32x32 pixel matrix using an interpolation algorithm to draw the lines.

In the case of an LSTM architecture, we use an LSTM layer and a dense layer of sizes 100 and 2, respectively. A softmax function was used as activation, and the categorical cross-entropy was utilized as an error function. A 0.2 dropout was used after the LSTM layer. We used pad sequences in trajectories with a maximum length of 500. Furthermore, the latitude, longitude, speed, and course were used in each sequence. The LSTM architecture can be seen in figure 5. In the NN, CNN, and LSTM models, we set the number of epochs to 100.

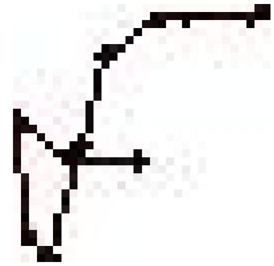

Fig. 4: AIS points of vessel in a fishing trajectory converted to a 32x32 pixel matrix using an interpolation algorithm to draw the lines.

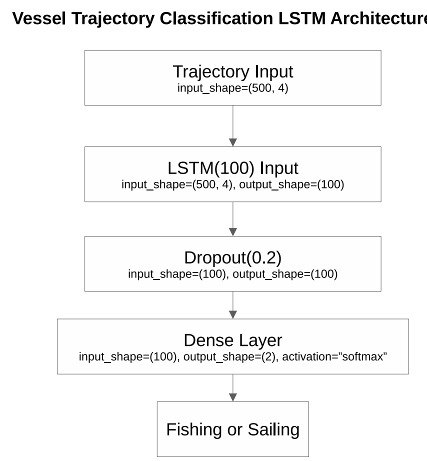

Fig. 5: The vessel trajectory classification LSTM architecture.

In the experiments, we used a cross-validation technique to better evaluate a model's generalizability. The technique contains a single parameter called k that specifies the number of groups into which a given data sample must be divided. Each group is made by preserving the percentage of samples for each class. it is a useful approach since it is intuitive and produces a less biased model or an optimistic evaluation of the model's performance. We randomly shuffled the data set and divided it into k = 5 parts. In training, each k was utilized as validation, and the best weights were stored to be used in the test.

We evaluated the results using assessment measures, namely precision, recall, and f-score (i.e., F1) [25]. Precision measures the number of times your model is correct in relation to the total number of times it tries to be correct, like in the equation 2. Recall measures the number of times your model is correct versus the total number of times it should have been correct, as can be seen in the equation 3. Finally, the F1 score is a metric that balances precision and recall, as in the equation 4. For example, if a model has high precision and predicts a positive instance, it is highly likely to be correct. In other words, the model has a low false-positive rate (FPR). However, if a model has a low recall, it indicates that it is failing to identify a significant number of relevant instances or that it has a high false-negative rate (FNR). This means that there are a significant number of instances that are positive, but the model is predicting them as negative. This way, we can see the precision, recall, and F1-score equations below:

The Precision equation:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP) + False Positives (FP)}} \tag{2}$$

The Recall equation:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP) + False Negatives (FN)}} \tag{3}$$

The F1-Score equation:

$$\text{F1} = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \tag{4}$$

Finally, we measure the execution time of the test samples' predictions. Each test sample had 1,468 trajectories for each class (fishing and sailing). The experiments used an Intel(R) Xeon(R) 3.30GHz processor with 48 cores, 256 GB of RAM, and two Nvidia Quadro RTX 8000 graphics cards.

IV. Results and Discussion

In this section, we will present the experiments results and make some remarks about the obtained data. In table I shows the model results for each pre-processing approach (trajectory-based (TB) and raw-based (RB)). The precision, recall, and f1-score metrics are shown for each class: fishing and sailing. A time column shows the execution time in seconds to predict the test sample data. The test sample data is composed of 1,468 fishing trajectories and 1,468 sailing trajectories.

The LSTM model has the highest precision and recall in the table I. The LSTM model fits very well with different trajectory sizes using the raw data. This means that the LSTM model has a high hit rate for both true positives (detection of a fishing trajectory) and true negatives. The second-best model was RF with a 0.87 precision; Furthermore, the model presented a similar recall using the trajectory-based data. This signifies that the model has a good hit rate for true positives and true negatives. A noteworthy fact is that, despite the LSTM's high accuracy, it required a significant amount of processing time. To predict a test sample data set, the LSTM model was 1,000 times slower than the second-best model.

In the table II we can see the train time performance and accuracy of the models. it is possible to note that the LSTM model has the slowest time to train, 2,200 times slower than the RF model. The fastest model to train was the LR, with only 8.28 seconds, but with an accuracy of 0.85. Another point to be noted is that the second best train accuracy was NN with 0.92. This can be explained by the high recall rate of 0.94 and the low precision rate of 0.60.

The LSTM model showed much better performance than other models because of its capability to deal with time series problems, mainly in cases when we compare a long trajectory with a short trajectory. Long trajectories tend to have a high variance in speed and course trajectory, penalizing the strategy to use this data in the models. The same situation can occur in the CNN model when the trajectories are transformed into images; the model will compare a long trajectory in an image with a short trajectory without taking into account the trajectory's duration and length.

The performance of the LSTM architecture may have been enhanced by the use of latitude and longitude data, which were unavailable in trajectory-based approaches. Due to the dispersed nature of the ship trajectories, this location characteristic may have been used to identify fishing vessels that frequent nearby fishing spots. This characteristic cannot be detected by the solution based on trajectory data.

Hence, we can conclude that the LSTM model is the most accurate model but, with a high demand for processing and the being the slowest model. Finally, the second-best model in terms of precision was RF using the trajectory-based data approach. The model presented good precision with a low processing time, but not as much precision as an LSTM model.

V. Conclusion

In a naval inspection scenario where maritime authorities must be certain that a vessel is on a fishing trajectory, this study compared the performance of the major machine learning approaches in the literature on the same data set to detect fishing activity in vessel trajectories. To meet this scenario, we need a model with high precision, i.e., with high true positive rates. To find the best solution, we compared a trajectory-based pre-processing approach and a raw data approach to use in the machine learning models, like LR, DT, SVM, RF, NN, CNN, and LSTM, measuring the accuracy and time performance for each one.

The LSTM model has 100% accuracy in detecting fishing activity in vessel trajectories using the GFW dataset and non-fishing vessels dataset. The LSTM architecture demonstrates that it can effectively compare trajectories of different lengths, which other approaches cannot do. Another factor that may have contributed to the high accuracy of the LSTM architecture is the sensitivity of the latitude and longitude data, which may have considered the proximity of fishing vessel spots. This is a characteristic that cannot be detected by the trajectory-based approach. However, the LSTM model is the slowest model at predicting the test samples. The second-best precision model is the RF model using trajectory-based data in the pre-processing, with 87% accuracy but 1,000 times faster than the LSTM.

We observe a trade-off between computational cost (time) and precision. In a scenario of IoT deployment, where models may need to process data in real-time or near real-time, the time performance of the model can be a critical factor. Despite this, in the case of this specific application, the time required for the most accurate model is permissible because the cost of an error is high. Thus, in a naval inspection situation where several resources need to be allocated, like navy ships, crew,

		Fishing Class					Sailing Class				
	Model	***Precision***	***Recall***	***F1***	***Accuracy***	***Time (sec.)***	***Precision***	***Recall***	***F1***	***Accuracy***	***Time (sec.)***
TB	LR	0.84	0.80	0.82	0.82	0.0023	0.79	0.83	0.81	0.82	0.0023
	DT	0.85	0.84	0.84	0.84	**0.0021**	0.84	0.85	0.84	0.84	**0.0021**
	SVM	0.77	0.81	0.79	0.79	0.2154	0.82	0.78	0.80	0.79	0.2154
	RF	0.87	0.84	0.85	0.85	0.0153	0.84	0.87	0.85	0.85	0.0153
	NN	0.60	0.94	0.73	0.78	0.0352	0.96	0.71	0.82	0.78	0.0352
RB	CNN	0.82	0.77	0.79	0.78	1.2761	0.75	0.80	0.77	0.78	1.2761
	LSTM	**1.00**	**1.00**	**1.00**	**1.00**	15.3384	**1.00**	**1.00**	**1.00**	**1.00**	15.3384

TABLE I: Table results show each model's precision, recall, F1, accuracy, and prediction time. The time column represents the execution time in seconds for 1,468 predictions per class in the test sample.

	Model	Train Accuracy	Prediction Time	Train Time
TB	LR	0.85	0.0023	**8.29**
	DT	0.87	**0.0021**	29.11
	SVM	0.83	0.2154	9,431.81
	RF	0.88	0.0153	17.37
	NN	0.92	0.0352	155.62
RB	CNN	0.78	1.2761	3,284.60
	LSTM	**1.00**	15.3384	38,438.27

TABLE II: The table shows the train accuracy of each model. The prediction time is the execution time in seconds for 1,468 predictions per class in the test sample. The column train time shows the time in seconds to train the model for 100 epochs.

environmental inspectors, drones, and others, a false positive can be expensive. In this situation, it is worth investing in high processing capability to run an LSTM model monitoring thousands of vessels.

REFERENCES

[1] D. Agnew, J. Pearce, P. Ganapathiraju, T. Peatman, R. Watson, J. Beddington, and T. Pitcher, "Estimating the worldwide extent of illegal fishing," *PloS one*, vol. 4, p. e4570, 02 2009.

[2] "Solas convention," https://www.imo.org/en/KnowledgeCentre/ConferencesMeetings/\-Pages/SOLAS.aspx, [Online; accessed 07-March-2023].

[3] I. Kontopoulos, A. Makris, and K. Tserpes, "A deep learning streaming methodology for trajectory classification," *ISPRS International Journal of Geo-Information*, vol. 10, 4 2021.

[4] S. Arasteh, M. A. Tayebi, Z. Zohrevand, U. Glässer, A. Y. Shahir, P. Saeedi, and H. Wehn, "Fishing vessels activity detection from longitudinal ais data." Association for Computing Machinery, 11 2020, pp. 347–356.

[5] Z. Zhang, L. Huang, X. Peng, Y. Wen, and L. Song, "Loitering behavior detection and classification of vessel movements based on trajectory shape and convolutional neural networks," *Ocean Engineering*, vol. 258, 8 2022.

[6] M. D. Ferreira, G. Spadon, A. Soares, and S. Matwin, "A semi-supervised methodology for fishing activity detection using the geometry behind the trajectory of multiple vessels," *Sensors*, vol. 22, 8 2022.

[7] L. Zhao and G. Shi, "Maritime anomaly detection using densitybased clustering and recurrent neural network," *The Journal of Navigation*, vol. 72, no. 4, 2019.

[8] D. S. Pedroche, D. Amigo, J. García, and J. M. Molina, "Architecture for trajectory-based fishing ship classification with ais data," *Sensors (Switzerland)*, vol. 20, pp. 1–21, 7 2020.

[9] K. Sheng, Z. Liu, D. Zhou, A. He, and C. Feng, "Research on ship classification based on trajectory features," *Journal of Navigation*, vol. 71, pp. 100–116, 1 2018.

[10] GFW, "Global Fishing Watch," https://globalfishingwatch.org/, 2023, [Online; accessed 07-March-2023].

[11] G. F. Watch, "Global Fishing Watch Dataset," https://globalfishingwatch.org/data-download/datasets/public-training-data-v1, 2023, [Online; accessed 07-March-2023].

[12] V. D. do Nascimento, "A Performance Analysis of Machine Learning Models for Fishing Activity Detection in Vessel Trajectories in a Maritime Authority Inspection Scenario Dataset," https://github.com/vindn/fishing_activity_detection/blob/main/dataset/dataset_fishing_train.csv.zip, 2023, [Online; accessed 07-March-2023].

[13] Oceana, "Oceana," https://oceana.org/, 2023, accessed: 2023-03-08.

[14] SkyTruth, "Skytruth," https://skytruth.org/, 2023, accessed: 2022-03-08.

[15] Wikipedia, "Global Fishing Watch," https://en.wikipedia.org/wiki/Global_Fishing_Watch, 2023, [Online; accessed 07-March-2023].

[16] A. Androjna, M. Perkovic, I. Pavic, and J. Miskovic, "Ais data vulnerability indicated by a spoofing case-study," *Applied Sciences (Switzerland)*, vol. 11, 6 2021.

[17] ITU-R, "Technical characteristics for an automatic identification system using time-division multiple access in the vhf maritime mobile band," *International Telecommunication Union, Recommendation ITU-R M.1371-5*, 2014.

[18] W. McKinney, "pandas: a foundational python library for data analysis and manipulation," *Python for High Performance and Scientific Computing*, vol. 14, pp. 1–9, 2011.

[19] G. D. Team, "Geopandas: Python tools for geographic data," *Journal of Open Source Software*, vol. 3, no. 24, p. 3522, 2018.

[20] A. Graser, "Movingpandas: Efficient structures for movement data in python," https://github.com/anitagraser/movingpandas, 2021, accessed: 2023-03-08.

[21] V. D. do Nascimento, "A Performance Analysis of Machine Learning Models for Fishing Activity Detection in Vessel Trajectories in a Maritime Authority Inspection Scenario," https://github.com/vindn/fishing_activity_detection, 2023, [Online; accessed 07-March-2023].

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[24] F. L. Gaol, "Bresenham algorithm: Implementation and analysis in raster shape." *J. Comput.*, vol. 8, no. 1, pp. 69–78, 2013.

[25] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation," in *International Conference on Machine Learning and Applications*. IEEE, 2011.