Ensemble Learning Approaches for Detecting Fishing Activity in Maritime Surveillance: A Performance Evaluation

Vinicius D. do Nascimento, Claudio M. de Farias and Diego L. C. Dutra
PESC/COPPE
Federal University of Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil
dalto@cos.ufrj.br, cmicelifarias@os.ufrj.br, ddutra@cos.ufrj.br

Tiago A. O. Alves
IME
State University of Rio de Janeiro (UERJ)
Rio de Janeiro, Brazil
tiago@ime.uerj.br

***Abstract*—Detecting fishing trajectories in maritime surveillance is of the utmost importance for identifying illegal fishing activity. In the event of illegal fishing activity, the maritime authority can mobilize resources to engage the vessel; hence, a false flag can be costly. This study investigates the efficacy of ensemble learning techniques for boosting individual model performance and decreasing uncertainty. Employing a range of machine learning models, including logistic regression, decision trees, random forests, neural networks, gradient boosting, and recurrent neural networks, the research evaluates the combination of these using ensemble methods like ensemble mean, weighted ensemble, and stacking approaches to enhance precision and decrease uncertainty. The primary dataset comprises a combination of fishing vessel and cargo vessel trajectories to train and test the models. Methodologically, the paper details the process of data analysis and the application of ensemble learning. A comparative assessment of individual models versus ensemble techniques forms the crux of this study. Results indicate a marked improvement in accuracy and consistency when employing ensemble methods, with weighted and stacking ensembles showing particular promise. These findings suggest that ensemble models outperform their individual counterparts in the context of maritime surveillance. This research makes a notable contribution to the maritime surveillance domain, demonstrating the potential of ensemble learning in enhancing detection capabilities for illegal fishing activities. The implications of these advancements are critical for maritime authorities as they strive to effectively monitor and protect marine ecosystems.**

***Index Terms*—fishing, trajectory, ensemble, illegal, performance**

I. Introduction

Illegal, unreported, and unregulated (IUU) fishing seriously jeopardizes the biodiversity of marine ecosystems. One example is the phenomenon of overfishing, which has the potential to exhaust fish populations, disturb the intricate web of food chains, and modify the equilibrium of species within an ecosystem. IUU fishing techniques, such as the method of bottom trawling, have the potential to do significant harm to coral reefs and seafloor habitats. These habitats play a vital role in the survival and reproductive processes of numerous marine species. One strategy to combat illegal fishing is by analyzing vessel behavior and the areas where fishing occurs.

Through Maritime Situational Awareness (MSA), it's possible to improve coordination between the maritime community and authorities, assisting maritime surveillance analysts in identifying suspicious activities. MSA significantly depends on sensors, with AIS being the most common sensor used on ships today. The Safety of Life at Sea (SOLAS) convention [1], issued by the International Maritime Organization (IMO) and verified by Maritime Authorities (MA), mandates all ships over 300 gross tons to carry an AIS transponder.

A skilled human operator can distinguish between images by identifying vessels in sailing or fishing engagement, checking adherence to an optimal route, bathymetric circumstances, meteorological elements, and other vessels that could endanger navigation. Examining thousands of images in a short time makes this process impractical. Thus, maritime situational awareness systems need automated classifiers. In the case of a fishing trajectory detection inside a forbidden fishing zone in the maritime authority monitoring system, which may require a vessel inspection, it is imperative that the alert possess a high level of precision and reliability. The MA is responsible for allocating a range of resources for conducting inspections. These resources include specialists such as environmental and labor inspectors, federal revenue officers, and anti-drug inspectors. Additionally, material resources like drones and helicopters can be utilized. Furthermore, specific criteria are considered when selecting the vessel designated for inspection, including factors such as maximum speed, type of radar, storage capacity, and weaponry capabilities. Hence, the occurrence of a false positive could potentially incur significant costs for the MA.

Fishing trajectory detection has been addressed by several works employing AIS sensor data in machine learning models. In [2] and [3] utilized AIS data to reconstruct a vessel's trajectory in an image, which was then employed as input in a Convolutional Neural Network (CNN) architecture. Similarly, [4] used CNN to classify vessel trajectories, while [5] employed AIS raw data to create a semi-supervised method for labeling vessel trajectories and a Recurrent Neural Network (RNN) architecture for classifying the trajectories into fishing and sailing. Other works, like [6], utilized RNN as a classifier for distinguishing fishing and non-fishing vessel trajectories. For the vessel trajectory classification problem, the predominant architecture used in RNN-based literature is the long short-term memory (LSTM). Hence, in this paper, we will consistently refer to the RNN model as using an LSTM architecture. Unlike previous works that used raw AIS data directly, in [7] performed preprocessing on the data, transforming AIS data into trajectories, and then utilizing the trajectory-based data as input in the models. This implies that aggregated data from trajectories, such as average speed and course variance, can be employed as inputs in machine learning models. For the issue of detecting fishing trajectories, this approach may be beneficial, given that variations in speed and course are distinctive. A similar approach was adopted in [8], using trajectory-based data in a logistic regression (LR) model to classify vessel trajectories. In [9], a performance comparison was conducted among solutions for detecting fishing activities using vessel trajectory data. However, the solution employing RNN exhibited superior performance compared to other solutions due to the utilization of geolocation data. This model was particularly favored because vessels that sail close to fishing locations can be considered fishing vessels by the RNN. Therefore, in this work, we will not use latitude and longitude data in the models; instead, we will solely use movement data such as SOG (Speed Over Ground), COG (Course Over Ground), and timestamps for a fairer comparison. Additionally, we will introduce other models like gradient boosting and employ ensemble techniques to enhance performance. We will not use SVM (Support Vector Machine) and CNN models because, in [9], these models present low performance achieving an accuracy close to 78%.

The significant contribution of this work lies in presenting a performance comparison among ensemble learning solutions for detecting fishing activity in vessel trajectories, trying to reduce uncertainty towards a reliable model. Ensemble strategies are potent approaches in the machine learning domain, notable for their capability to produce models that are more accurate, robust, and generalizable than individual models. These techniques function by combining the predictions from multiple models, which assists in reducing both bias and variance, thereby mitigating the risks of overfitting and underfitting and reducing the risk associated with selecting an unsuitable model. Moreover, ensembles are particularly adept at handling noisy data and outliers, leveraging the "wisdom of the crowd" to achieve a more comprehensive and precise understanding of the data. The flexibility to combine different model types allows ensembles to adapt to a broad range of tasks and data sets, exploring multiple hypotheses about the data structure.

Despite their increased computational complexity and the challenges in interpreting results, the benefits provided by ensemble strategies justify their utilization. They not only enhance the accuracy of predictions but also bolster the robustness of models against variations in training data. In order to achieve our goals, we will conduct a comparative analysis of various methodologies and models, employing a standardized dataset, and assess their predictive accuracy through the examination of test data samples. The dataset provided by Global Fishing Watch [10] will be combined with another dataset containing information on non-fishing vessels in transit. The objective of this fusion is to effectively distinguish between trajectories that involve fishing activities and those that do not. The dataset utilized in this study can be accessed from the source provided in the reference [11]. The GFW was established in September 2016 through a partnership between Google, Oceana, and SkyTruth. It offers the first comprehensive worldwide analysis of commercial fishing operations.

Thus, one of the objectives of this work is to investigate a model enhancement that decreases uncertainty in detecting a fishing trajectory and achieves a high true positive rate, facilitating the identification of illegal fishing using vessel trajectory as the input. In the wake of this condensed introduction, the organization of this work develops as follows:

- A methodology section elucidating what vessel trajectories include, their construction methodology, the data used, preprocessing, the approaches and models adopted, the primary strategies for detecting fishing activity, and the ensemble learning methodology used in this work;
- An experiments section describes the model parameters used in the experiments, the metrics, the data set division, and the validation strategy;
- A section on results and discussion in which the authors reflect on the most significant findings and explain the factors that contributed to them; and
- And a conclusion section accentuating the most notable accomplishments of this work.

II. METHODOLOGY

In pursuit of our primary objective, namely the reduction of uncertainty in model-based detection of fishing trajectories, our initial step involves the implementation of individual models dedicated to this end. These models are delineated in [9], with their source code accessible via GitHub [12]. Beyond the scope of the models presented in this referenced work, our study incorporates the gradient boosting model. The decision to incorporate this model was based on its distinct characteristics compared to other models, which operate by constructing models sequentially, where each new model corrects errors made by previous ones. The differentiation in the characteristics of how models are constructed is crucial in the use of ensembles, as one model can complement the deficiencies of another. Aiming to enhance the efficacy of the individual models cited in [9], modifications were applied to the original code. A pivotal alteration is the exclusion of geolocation data, specifically latitude and longitude, from our analysis. The inclusion of such data in the aforementioned study inadvertently benefited the RNN model, as it tended to classify vessels nearing fishing zones as fishing vessels. While this approach may be viable for monitoring fixed regions, it diverges from our objective of identifying fishing trajectories irrespective of geographic coordinates. Consequently, the parameters selected for training and inference in our models are Course Over Ground (COG), Speed Over Ground (SOG), and timestamp data. Furthermore, we introduce a novel approach to processing the COG dimension, employing circular direction difference calculations to ascertain the minimal angular deviation rather than a mere differential COG value. In the realm of RNN model enhancement, besides the inclusion of SOG and COG dimensions, we introduce a dimension representing the angular disparity between consecutive points and another dimension capturing the temporal interval between these points. The angular difference dimension introduces the extent to which the vessel has changed direction from one point to another, while the temporal interval dimension highlights the time interval between the transmission of AIS messages, potentially refining the characterization of the fishing trajectory. After the implementation of these individual model modifications, our subsequent phase entails the training and inferential analysis of each model to evaluate their respective performances. This analysis will guide the formulation of an ensemble model strategy predicated on the individual models' efficacy. In summation, this section elucidates the concept of a fishing trajectory, outlines detection methodologies, details the models employed, and discusses the ensemble strategies.

A. Vessel Fishing Trajectories

In the quest to maximize profitability, maritime navigation strategies are optimized to conserve fuel consumption and avoid routes with navigational hazards [13]. Each vessel is designed with a unique purpose, though the majority are intended for the transportation of goods or transit between an origin and a destination. The trajectory of cargo ships is predominantly linear, aiming to optimize the efficiency of the route.

Fishing vessels behave similarly to cargo ships in their routine transit, likewise striving for optimized navigation. Upon entering fishing regions and commencing their primary operation, these ships execute specific maneuvers tailored to their fishing technique. These maneuvers exhibit various changes in direction and speed, depending on the type of fishing being conducted. In Figure 1, we can observe a fishing trajectory on the left side, where we have many changes of direction and speed.

The composition of ship trajectories comprises a set of Automatic Identification System (AIS) messages, each containing details of the vessel's location (latitude and longitude), speed (in knots), direction (in degrees), as well as specific vessel information (name, maritime mobile service identity (MMSI), International Maritime Organization (IMO) number, and IRIN), and voyage data (origin, destination, and status). It is crucial to note that data such as origin, destination, and status are manually entered by the crew, while coordinates, direction, and speed are automatically obtained from GPS devices. Manual entries are susceptible to inaccuracies,

Fig. 1: Example of vessel trajectories.

whether intentional or inadvertent. Although this issue has been addressed in other studies [14], the present study assumes the reliability of the information provided in the dataset, except for the correction of impossible positions and speeds. In Figure 1, we can observe examples of trajectories; on the left side, there is a fishing trajectory, while on the right side, there is a trajectory of a ship merely navigating towards a destination. The points represent the AIS messages sent by the ships at a given moment, informing their latitude, longitude, speed, and direction.

B. Approaches for Detecting Fishing Trajectory

The Automatic Identification System (AIS) is collaborative equipment installed on vessels, reporting information about position, identity, and the origin and destination of trajectories [15]. AIS messages sent by vessels are received by coastal antennas installed by the Vessel Traffic Management System (VTMS), maritime companies, or governmental authorities. Another method of receiving these messages is through low-orbit satellites used by some private companies, which collect the data and subsequently sell it. This AIS message is typically utilized in Maritime Situation Awareness (MSA) systems by private enterprises or government agencies. Certain AIS datasets are available to academic researchers, such as those from Global Fishing Watch (GFW) [10]. GFW is an online platform providing almost real-time views of global fishing activities, utilizing satellite data and vessel tracking.

Within the AIS dataset available on GFW [10], there are AIS messages from fishing vessels on fishing trajectories. We utilized this dataset from GFW combined with other AIS messages from non-fishing vessels found in [12]. In total, the dataset comprises about 2 million lines. While this may seem small for an AIS dataset, manually determining which trajectories are related to fishing is a challenging task.

In the dataset, each line contains data on location (latitude and longitude), identity (MMSI and vessel name), vessel speed and course, and a timestamp. A trajectory can be constructed by linking a set of AIS data from the same vessel. However, if we connect all points of a vessel into a single trajectory, we would be including stop points, which is not desired. Therefore, we will consider an interval between trajectories of the same vessel; that is, when there is a time difference greater than 90 minutes between two points, the trajectory will be divided into two. This approach results in different trajectories for the same vessel, allowing for the analysis of trajectory behavior in a specific location. Consequently, as we are interested in the movements of the vessels, we will only consider trajectories with average speeds between 1 and 50 knots and durations longer than 10 minutes.

To create these trajectories, we first generated data frames in Python using Pandas [16], and then transformed these data frames into geodata frames using GeoPandas [17]. GeoPandas is a Python library utilized for working with geospatial data. It is built upon Pandas and extends its capabilities by adding support for spatial data types and spatial operations. Subsequently, we employed these geodata frames to construct trajectories in Moving Pandas [18]. Moving Pandas is a Python library for the manipulation of movement data, based on Pandas and GeoPandas. Utilizing moving pandas, it is feasible to aggregate AIS points into trajectories, applying metrics such as average speed and course, variance in speed and course, trajectory duration, and so forth.

This strategy was employed in both raw data-based and trajectory-based approaches. However, only the trajectory-based approach, as described in [7], utilized aggregated trajectory-related data (for example, velocity 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 used in the model. In our case, we will use only speed, course, and timestamp data so that our detection can be performed in any location.

Aggregated trajectory data are not obvious data points that can be directly used in models. In Figure 1, we can observe that fishing vessels tend to have more course variations. Therefore, if we calculate the course and speed variance of the AIS points in trajectories, we can use this metric in our models. To calculate the variance between angles, we need to compute it in a circular manner, so that if we have one direction at 355 degrees and another at 10 degrees, the correct difference between the angles is 15 degrees, not 345 degrees. For this purpose, we will calculate the angular difference following the equation 1.

$$\Delta\theta = \arctan 2(\sin(\theta_1 - \theta_2), \cos(\theta_1 - \theta_2)) \tag{1}$$

Where θ_1 and θ_2 are the directions at each point, and $\arctan 2$ is a variation of the arc tangent function that takes into account the sign of both the sine and cosine components to determine the quadrant of the resulting angle.

Although it may seem like a straightforward method to differentiate one trajectory from another using variance, there are other variables that can cause an increase in variance. If a vessel has a long trajectory, the variance tends to increase. In an optimized path, a vessel may navigate around the coast of a country, which can increase the course variance. Therefore, the duration and length of the trajectory can interfere with the variance. Another variable to consider is the number of points in a trajectory. In our case, we will consider a minimum of three points for a trajectory so that we can better observe the vessel's direction variance. Similarly, we can use the vessel's speed variance.

Thus, we employed two strategies for preprocessing the AIS data of each ship. The first uses aggregated data based on the trajectory, and the second uses raw data with the time series. Consequently, we have two approaches:

- Trajectory-based aggregated data dimensions (for models that do not use time series):
  - Trajectory duration: the total time of the segment;
  - Angular variance: the calculated variance of the vessel's course in the trajectory;
  - Speed variance: the calculated variance of the vessel's speed in the trajectory;
  - Trajectory length: the length of the trajectory in meters;
  - Number of points: number of points in the trajectory;
- Raw data dimensions based on time series (for models that use time series):
  - Speed;
  - Course;
  - Time difference between points;
  - Angular difference between points;

For a fair comparison, we used the same trajectories in both preprocessing approaches, with the strategy dependent on the model used. For instance, if we use logistic regression as a model, we would employ the first preprocessing strategy with aggregated data. In the case of using an RNN, the second preprocessing strategy would be utilized, using raw data with the time series. In the following sections, we will detail the models and parameters used.

C. Ensemble Learning Approach

Ensemble learning is a machine learning technique that involves combining multiple models to enhance accuracy and performance in solving complex problems. The core idea behind ensemble learning is that by amalgamating the predictions of multiple models, it is possible to achieve more precise and robust outcomes than any single model could provide. Historically, the concept of ensemble learning began to gain prominence in the 1990s, with the development of algorithms such as bagging and boosting. These methods demonstrated that the aggregation of multiple learning models, each contributing its own perspective or expertise, can lead to significantly improved performance, particularly in complex classification and regression tasks. The basic premise is that while a single model may have its limitations and be prone to specific errors, the combination of several models can offset these weaknesses, resulting in greater accuracy and reliability in predictions [19]. As our aim is to reduce uncertainties, this strategy will assist us by allowing each model's strengths to compensate for the weaknesses of others, generating more stable predictions over multiple iterations.

As individual models, we will utilize the following: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Neural Network (NN), Recurrent Neural Network, and Gradient Boosting (GB). The performance of these models in detecting fishing trajectories was observed in the work of [9]. However, we have made modifications to the dimensions that will be used in the individual models, resulting in altered model performances.

With the aim of enhancing the performance of predictions, we will employ ensemble methods strategies to combine the outputs of models to obtain a more robust model, thereby reducing uncertainties in detecting fishing trajectories. As a strategy to combine the outputs of the models, we will use ensemble mean, ensemble weighted, and stacking methods. Thus, we can describe the following characteristics of each method:

- Ensemble Mean - Predictions from all models in the ensemble are aggregated through the calculation of the arithmetic mean to obtain the final prediction. This technique is commonly used in situations where an equivalent reliability is assumed among all the constituent models of the ensemble. For example, in the presence of three distinct models making predictions for the same task, the procedure involves calculating the arithmetic mean of the predictions provided by each of these models. The effectiveness of this method is particularly evident in contexts where individual models are diverse and operate independently, as the mean helps to mitigate extreme predictions and reduce variance.
- Ensemble Weighted - Each model in the ensemble is assigned a distinct weight, which is commonly determined based on the performance or accuracy of each model. In this context, the combination of predictions from individual models is carried out considering the weights assigned to each. Models demonstrating superior performance exert a more significant influence on the final prediction. This approach is particularly advantageous in situations where certain models consistently exhibit higher accuracy compared to others. The use of weighting allows the ensemble to benefit from performance discrepancies among the various models, thus optimizing the overall accuracy of the predictions.
- Stacking - Represents an advanced approach within ensemble techniques, characterized by combining multiple classification or regression models. In stacking, predictions generated by a variety of models, called base models, are used as input variables for a new model, known as the meta-model or second-level model. This meta-model is specifically trained to make the final prediction. The efficacy of stacking is particularly notable in scenarios where there is significant heterogeneity among the base models. Such diversity allows the meta-model to develop an enhanced ability to discern the most efficient way to integrate individual predictions from the base models, resulting in superior accuracy in the final prediction.

In the next section, we will detail the parameters used in each model and ensemble in the experiments, as well as the architectures used in the machine learning models.

III. Experiments

Firstly, in our experiments, we will execute each model individually and evaluate its performance. This is done with the intention of selecting the best combinations of models to achieve improved accuracy in the ensembles. For each model, we will specify the parameters used in the following experiments.

In the dataset, we applied filters to remove impossible trajectories and trajectories where the vessels are stationary. For this, we used trajectories that have a minimum duration of 10 minutes and an average speed between 1 and 50 knots. After data cleaning and trajectory creation, there remained 7,336 fishing trajectories and 7,336 sailing ship trajectories, where 20% of the trajectories will be used for testing and 80% for training the models.

We used GridSearchCV from the scikit-learn library [20], dividing the dataset into 5 parts. We employed the data preprocessing strategy based on aggregated trajectory data [7] and searched for the best parameters in the following models:

- Logistic Regression (LR): The best parameter values were C=0.01 and regularization='l1';
- Decision Tree (DT): The best parameters were criterion: entropy, max_depth: 10, and number of components: 5;
- Random Forest (RF): The best parameter values were criterion: gini, max_depth: 5, n_estimators: 150;
- Gradient Boosting (GB): The best parameter values were learning_rate: 0.1, max_depth: 3, min_samples_leaf: 1, min_samples_split: 2, n_estimators: 300, and subsample: 0.9;
- Neural Network (NN): 4 dense layers (32, 16, 8, and 2) were used, with softmax activation function, categorical cross-entropy loss function, and rmsprop optimizer;

In the case of the RNN model, we used raw data with the time series to train the model. We employed a LSTM layer of size 100 and a dense layer of size 2. The loss function used was categorical cross-entropy, and the optimizer was rmsprop.

Regarding the ensemble models, we used the combination methods of ensemble mean, ensemble weighted mean, and stacking. In all three methods, we combined the models according to their individual performance 2, prioritizing models that achieved better precision and also considering the correlation matrix in table I.

In figure 2, we can observe the accuracy of the individual models over 50 rounds. During each round, the dataset is randomized, and training and test data are selected randomly. The graph shows the average accuracy over the 50 rounds as well as the standard deviation. In this case, we can highlight the models with the best accuracy, GB and RNN.

In the case of figure 3, we can observe the performance of precision, recall, and F1 score of the models. For our problem, it is important to have low false positives, meaning that what interests us has a higher precision. In this case, we can verify that the models that achieved the highest average precision were GB, RF, and RNN, respectively. However, in some instances, the NN reached a precision above 0.95, making it interesting to combine it with other models and study its behavior over the rounds.

In table I, we can see the correlation matrix between the predictions made by the models. Each value in the matrix represents how much one model agreed with another, with zero indicating no agreement and one hundred indicating

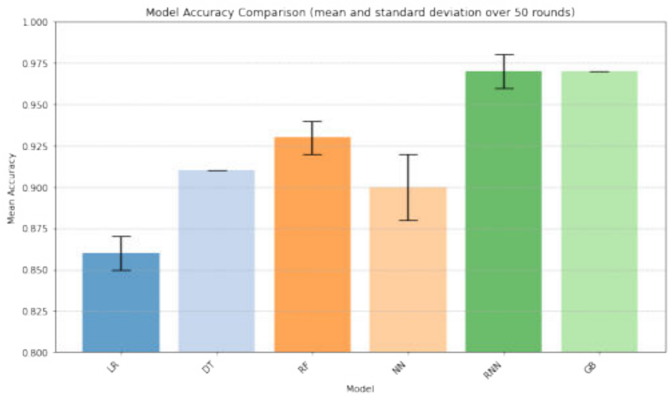

Fig. 2: Model accuracy of individual models.

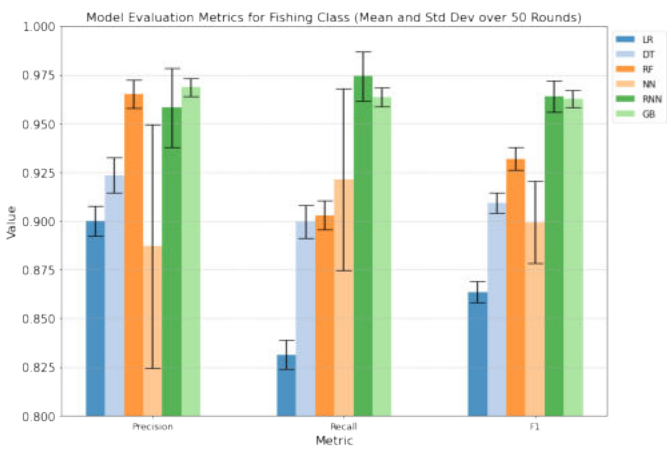

Fig. 3: Precision, recall and F1 performance of individual models.

	LR	DT	RF	NN	RNN	GB
LR	100.00	48.61	51.58	61.97	31.58	46.39
DT	48.61	100.00	58.34	50.14	41.55	54.85
RF	51.58	58.34	100.00	49.18	35.67	78.05
NN	61.97	50.14	49.18	100.00	49.14	47.33
RNN	31.58	41.55	35.67	49.14	100.00	43.68
GB	46.39	54.85	78.05	47.33	43.68	100.00

TABLE I: Correlation matrix of model predictions. Red rectangles represent low correlation, and green rectangles represent high correlation.

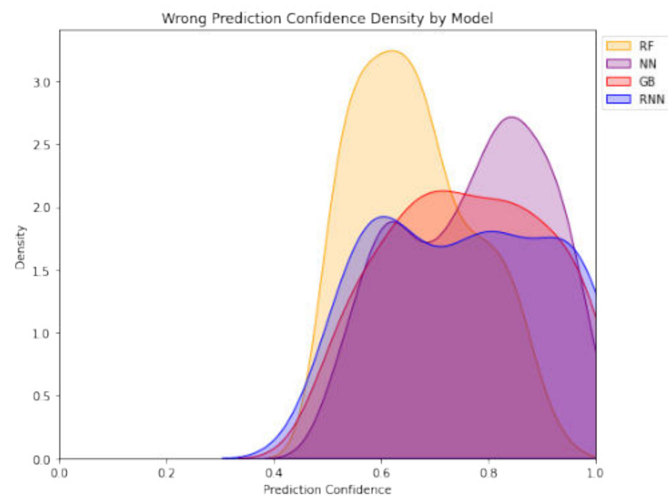

Fig. 4: Wrong prediction confidence density graphic. This graphic shows the distribution of fishing prediction values when the models miss.

complete agreement. The red rectangle in the matrix represents the lowest values. The highest values represented by green rectangle show the models that most agreed with each other, while the lowest values show the models that most disagreed with each other. Analyzing the correlation between the GB, RNN, and RF models, we can see that the GB and RNN models agreed only 43% of the time, however, the GB and RF models agreed about 78% of the time. When correlating the RNN model with the RF model, they agree about 35% of the time when predicting fishing. Therefore, we will give preference to the combination of models with higher precision in figure 3 and that diverged from each other in table I. In this case, we will combine the RNN and GB models, RNN and RF, and as the NN model in some rounds achieved the highest precision, we will combine them in the ensemble strategies. All models implemented here, along with the experiments, can be found in [11].

To define the weights of the models in the weighted ensemble, we will base our decision on the incorrect predictions of the individual models and assess how the confidence of one model can influence the others. For this, we can analyze figure 4, which represents the density of the prediction confidence of the models when an incorrect fishing prediction occurs. In the graph, we can see that the NN model, when making an incorrect fishing prediction, mostly does so with probability values greater than 0.8. In the case of the RF model, it mostly errs with values between 0.6 and 0.8. For the GB and RNN models, the values are distributed between 0.5 and 1.0. Therefore, we will consider that the NN model has a greater influence on the final prediction and the RF model has less influence. The RNN and GB models influence each other equally. Thus, to balance the final prediction value when combining the models, we should assign a higher weight to RF and a lower one to NN, while for the RNN and GB models, we should assign an intermediate weight.

In addition to the three types of ensembles we have presented, we will also test their performance with nested stacking models. This model follows the same idea as stacking, with the difference that a new level is created to combine the output of the first stacking with another model. In this case, we will combine the models with the highest individual performance.

IV. Results and Discussion

In table II, we can verify the results presented by the models over 50 rounds. The precision column represents the proportion between the trajectories classified as fishing that are indeed fishing (true positives) and the trajectories classified as

TABLE II: Comparison of models performance

Model	Precision	Recall	F1
LR	90.00	83.14	86.36
DT	92.34	89.96	90.92
RF	96.52	90.30	93.20
NN	88.70	92.14	89.94
RNN	95.82	97.44	96.40
GB	$\boxed{96.86}$	96.36	96.28
Ensemble Mean (RF, RNN, GB)	97.64	97.14	97.10
Ensemble Mean (RF, RNN)	97.24	97.36	96.98
Ensemble Mean (RNN, GB)	97.44	98.26	97.68
Ensemble Weighted (1RNN, 2GB)	97.24	97.26	97.04
Ensemble Weighted (2RNN, 1GB)	96.70	98.06	97.10
Ensemble Weighted (2RF, 1RNN)	$\boxed{97.82}$	95.80	96.58
Ensemble Weighted (2RF, 1RNN, 1GB)	97.68	96.44	96.80
Stacking (GB, RF)	96.54	96.50	96.26
Stacking (GB, RNN)	97.74	97.92	97.64
Stacking (RF, RNN, GB)	$\boxed{97.80}$	97.80	97.58
Stacking (RNN, RF)	97.30	97.66	97.20
Stacking(Stacking (GB, NN), RNN)	97.62	97.90	97.52
Stacking(Stacking (GB, RNN), NN)	92.80	91.54	91.94
Stacking(Stacking (RF, NN), RNN)	96.50	97.68	96.82
Stacking(Stacking (RF, RNN), NN)	97.28	97.38	97.00

fishing (true positives and false positives). The recall column represents the proportion between the trajectories classified as fishing that are indeed fishing (true positives) and all the trajectories that are actually fishing (true positives and false negatives). The F1 column is the harmonic mean of precision and recall, offering a balance between the two. In table II, we can observe that the weighted ensemble and mean ensemble models, which used the combination of RF, RNN, and GB models, mostly achieved better performance than the individual models. Notably, the ensemble weighted (2RF, 1RNN) model achieved the best precision of 97.82%. In the case of models using stacking, all also performed better than the individual models, with the Stacking (RF, RNN, GB) model presenting the second-best precision of 97.80%, slightly lower than the first. When evaluating different preprocessing techniques, it was revealed that models that utilized aggregation features obtained superior performance compared to those that used time series data. Nevertheless, the integration of these strategies in the ensemble methods produces the best results, which are shown in table II.

In figure 5, we can assess the consistency of the models over the rounds in terms of the precision metric. Models with smaller variations tend to be more consistent, i.e., those with a narrower whisker in the graph. The small diamonds in the graph represent outliers over the 50 rounds. We can see that the GB model showed low variation throughout the 50 rounds, maintaining precision at approximately 97% with some outliers. The ensemble models, such as Ensemble Weighted (2RF, 1RNN) and Stacking (RF, RNN, GB), showed little variation but between 97% and 99% and no outliers. Moreover, we can note that the NN model had high variation, but when combined with other models, there was a considerable decrease in its variation while still improving the precision of the set, as seen in the case of nested stacking models. Thus, we can verify that ensemble models can reduce the variation of the models, decreasing uncertainty when predicting a trajectory like fishing. This is particularly beneficial in our application, where the predictability of model performance is important.

It is also important to highlight that the stacking method can be influenced by the performance of the training of individual models. If an individual model suffers from overfitting, that is, its training has better accuracy than in testing, using the training data to train the meta-model will bias it towards the model that achieved higher accuracy in training.

V. Conclusion

In this study, we presented a comparison of different ensemble methods using individual models for the detection of fishing trajectories. These models proved to be suitable for applications that require precision while reducing the uncertainty of the ensemble.

We used two types of data transformations at the preprocessing level: the first using trajectory data aggregation and the second using time series. After transforming this data, we trained and tested the individual LR, DT, RF, NN, GB, and RNN models with the data. Then, after analyzing the individual models, we implemented and compared different ensemble techniques on the individual models, which increased the precision of the model and decreased the variation in precision over the rounds. The weighted ensemble method using RF and RNN models showed the best performance with a precision of 97.82%, and the second best was the stacking method using RF, RNN, and GB models, showing a precision of 97.80%. In an application where a false positive can be costly, as in the case of maritime authorities, where it may be necessary to send a navy ship to conduct an engagement, increasing the precision of the model while reducing uncertainty is essential.

Therefore, the use of ensemble models offers a promising approach for maritime surveillance, potentially enhancing the capabilities of authorities in monitoring fishing activities and thus contributing to sustainable fishing practices and ocean conservation.

Future research should focus on using data fusion techniques for these models with emerging technologies, such as satellite imagery and radars, to further increase their effectiveness in real-time applications.

Declaration of Generative AI

During the process of preparing this work, the authors utilized ChatGPT to enhance the linguistic quality and readability. Upon utilizing this service, the authors carefully reviewed and edited the content as required, assuming complete accountability for the publication's content.

References

[1] "Solas convention," https://www.imo.org/en/KnowledgeCentre/ConferencesMeetings/Pages/SOLAS.aspx, [Online; accessed 2023-12-18].

[2] 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.

[3] 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.

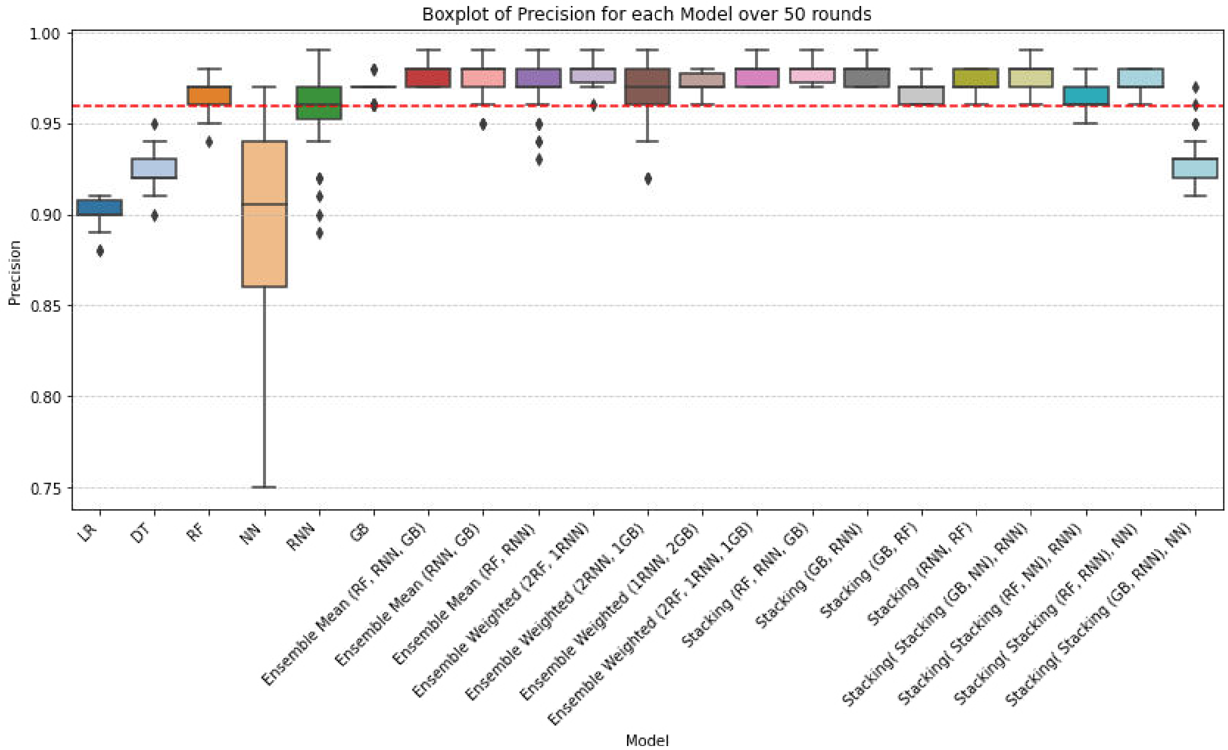

Fig. 5: Boxplot of precision for each model over 50 rounds. The red dot line is the mean of precision results from all models. In the boxplot, a short mustache is better because it represents fewer variations in performance in the runnings. The small diamonds represent the outliers.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] V. D. do Nascimento, T. A. O. Alves, D. L. C. Dutra, and S. Kundu, "Comparing approaches for detecting fishing activity in maritime surveillance: An evaluation of accuracy and performance," in *2023 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE'23) - The 25th International Conference on Artificial Intelligence (ICAI'23)*, Las Vegas, NV, 2023, p. To be published in IEEE proceedings. [Online]. Available: https://github.com/vindn/fishing_activity_detection/blob/main/IEEE_CPS_2023___Fishing_Activity_Detection__final_version_.pdf

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

[11] V. D. do Nascimento, "Source code of ensemble learning approaches for detecting fishing trajectories," https://github.com/vindn/ensemble_models_for_fishing_trajectory_detection/, 2024, [Online; accessed 05-January-2024].

[12] ——, "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] M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen, "Ship routing and scheduling in the new millennium," *European Journal of Operational Research*, vol. 228, no. 3, pp. 467–483, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221712009125

[14] 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.

[15] 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.

[16] 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.

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

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

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer Series in Statistics. Springer New York, 2009.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.