Identifying and Combating Unlawful Fishing Activities: A Classification-Based Approach

Maanaav Motiramani
B Tech Integrated,
Computer Engineering
MPSTME, NMIMS
Mumbai, India
maanaav9@gmail.com

Parth Mody
B Tech Integrated,
Computer Engineering
MPSTME, NMIMS
Mumbai, India
modyparth7@gmail.com

Param Sejpal
B Tech Integrated,
Computer Engineering
MPSTME, NMIMS
Mumbai, India
paramsejpal@gmail.com

***Abstract*—Illegal, unreported, and unregulated (IUU) fishing has become a very profitable business, with nearly $23.5 billion worth of fish stolen from the world's oceans. Every year fishing business grows efficiently, and Illegal fishing is one of the big back steps in this business. Furthermore, many fish populations are pushed to extinction. This paper proposes illegal fishing detection using data analytics and machine learning techniques. The primary data was gathered from the Global Fishing Watch (GFW) and the data was analyzed to find whether the vessels are used for illegal or legal fishing based on the sensors attached to the vessel GFW found location data, type of the vessel, and speed of the vessel. Even with the available data, trying to solve this problem globally is a very difficult task. By these classification models (Logistic Regression, SVM, KNN, Naïve Bayes, XG Boost, Decision Tree, Random Forest Classifier), we can predict illegal fishing and can take necessary actions against the illegal fishing boats.**

Keywords—Illegal, unregulated, and unlicensed fishing detection, Machine learning model, Classification, Global Fishing Watch

I. INTRODUCTION

Pirate fishing is also termed IUU (Illegal, unreported, and unregulated) fishing, expanded as illegal, unreported, and unregulated fishing. The process of collecting data from several trade figures, expert estimates, and fisheries control agencies is quite tedious and sometimes inaccurate. Miscalculations like assumptions of lower fish-catching quota, overestimating the size of the stock may lead to the determination of the following year's catch quota to be too high. This error may in turn lead to potentially entrenching and increasing the overexploitation of the stock as stated in. Though Unregulated fishing is not an offensive act under a nation's law, it is problematic in the cause of preserving biodiversity and keeping track of the fishing stock. Manual solutions come with deficits like insufficient and inadequately trained authority personnel, low financial importance, maintenance of patrol vessels, etc.

This research aims at machine learning models to handle this problem. The training data is downloaded from Global Fishing Watch (GFW). Global Fishing Watch is a partnership between Skytruth, Google and Oceana to map all the trackable commercial fishing activity in the world, in near-real time, and make it accessible to researchers, regulators, decision-makers, and the public.

The training dataset contains information of Kristina's trawler dataset. Kristina Boerder is one of the academic partners of GFW from Dalhousie University in Halifax, Nova Scotia. Trawlers for instance zig zag back and forth, moving at a constant speed, dragging nets behind them to collect their catch. Purse seiners, on the other hand, travel quickly in a tight circle, closing their nets around a school of fish in a matter of minutes. Longliner tracks are spikey, tracing the same line again and again as they set their hooks and return to retrieve their catch.

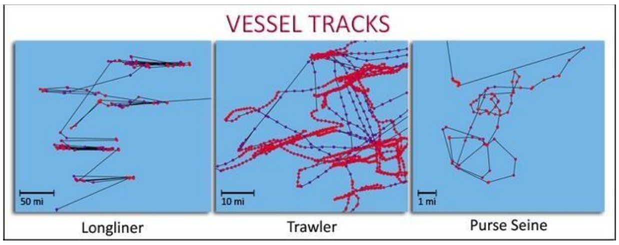


Fig. 1. Vessel tracks types

II. LITERATURE REVIEW

Even though illicit fishing has major negative effects that contribute to ecological and socioeconomic crises all over the world, data mining and machine learning experts have not given the problem enough attention. Below is described several works pertaining to illegal fishing detection using machine learning techniques.

The study "Illegal Fishing Detection Using Neural Network" puts forth a technique that attempts to classify the fishing activity of different vessels before using this to target suspicious actions in the open ocean [1]. The application of neural computing facilitates the processing of massive amounts of data generated by the sensors and handles numerous noise-related inconsistencies, saving time and money. To determine the illegal action that a vessel engages in, the authors V.K.G Kalaiselvi, et al. have combined geospatially referenced and physics-based sensor data.

In paper "Catching Illegal Fishing Using Random Forest and Linear Regression Models" used SAR (Synthetic Aperture Radar) satellite data collection to track and record information about each ship, including its location in the water and intended usage [2]. The authors B. Padmaja, et al. then implemented linear regression and random forest classifier to achieve accuracy of 84%.

The regression model is used in the work "Catching of Illegal Fishing using Data Analytics" to identify the behaviour of the vessel and establish whether it is fishing or not [3]. To pinpoint unlawful fishing, the authors B. Jyothi, et al concentrated on a particular geographic area. There are two main divisions in the project's meta-model. The first model is a regression model based on the vessel's type, speed, and AIS location data. By doing so, the authors were able to determine whether the vessel was fishing or not..

III. ARCHITECTURE

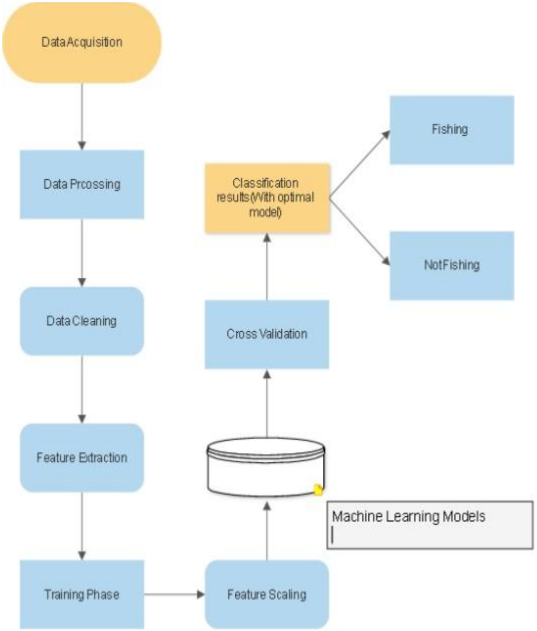
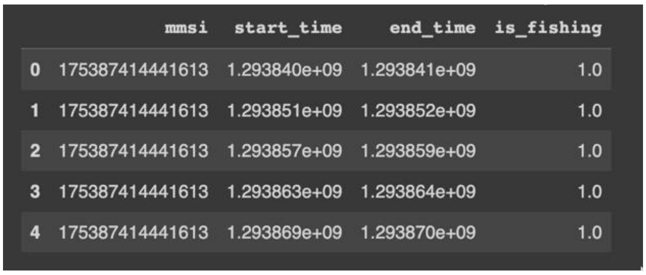


Fig. 2. Model architecture

1. Data acquisition is the process of acquiring data, the dataset for this problem was acquired from the Global Fishing Watch (GFW).
2. The next part was to clean the data i.e. (removing erroneous data and imputing missing values) so that it can befeed to the model.
3. Feature extraction and feature selection was performed extraction involves transforming the raw data into meaningful numeric data and selection involves choosing of appropriate features for making predictions and removing non-informative or redundant features from the model, one can use domain knowledge or methods like chi-square, co-relation matrix etc to determine the same.
4. The next part was the training phase, where the dataset was split into training data and testing data by an 80%-20% proportion, respectively.
5. In the subsequent phase to training, feature scaling was performed to standardise the values of feature across the dataset.
6. In the next step the model was built using multiple algorithms and was feed with the data prepared/pre-processed beforehand.
7. Cross-validation techniques were used to find the most optimal solution.
8. Predictions are made based on features/input data and classified as Fishing(1) or NotFishing(0).

IV. METHODOLOGY

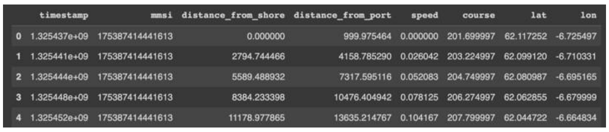
For data collection training set from Global Fishing Watch was downloaded. Two separate files were downloaded from their GitHub page. The first file contains time information for Kristina's trawler dataset. This data is the labelled data and contains information that for a given time the vessel was fishing or not. This dataset has information for 7 trawlers: 175387414441613, 269050323939773, 231154271004480, 143906914639303, 218796484670282, 274850145767759 and 222656062190286.



	mmsi	start_time	end_time	is_fishing
0	175387414441613	1.293840e+09	1.293841e+09	1.0
1	175387414441613	1.293851e+09	1.293852e+09	1.0
2	175387414441613	1.293857e+09	1.293859e+09	1.0
3	175387414441613	1.293863e+09	1.293864e+09	1.0
4	175387414441613	1.293869e+09	1.293870e+09	1.0

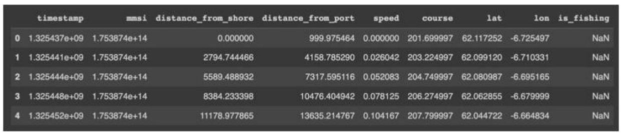
Fig. 3. Snippet of the Kristina's trawler dataset

The second file downloaded contained the track information for vessel of Maritime Mobile Service Identity (MMSI) 175387414441613. This file is in npz format. npz is a numpy file format that stores data in named variables. It contains the independent / input features timestamp, mmsi, distance from shore and port, speed of vessel and course along with the latitude and longitude of the vessel for a given time.



	timestamp	mmsi	distance_from_shore	distance_from_port	speed	course	lat	lon
0	1.325437e+09	175387414441613	0.000000	999.975464	0.000000	201.699997	62.117252	-6.725497
1	1.325441e+09	175387414441613	2794.744466	4158.785290	0.026042	203.224997	62.099120	-6.710331
2	1.325444e+09	175387414441613	5589.488932	7317.595116	0.052083	204.749997	62.080987	-6.695165
3	1.325448e+09	175387414441613	8384.233398	10476.404942	0.078125	206.274997	62.062855	-6.679999
4	1.325452e+09	175387414441613	11178.977865	13635.214767	0.104167	207.799997	62.044722	-6.664834

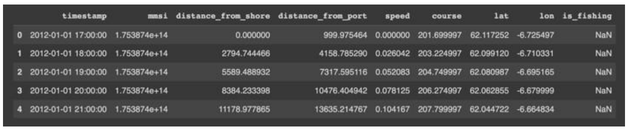
Fig. 4. Snippet of the Maritime Mobile Service Identity track dataset



	timestamp	mmsi	distance_from_shore	distance_from_port	speed	course	lat	lon	is_fishing
0	1.325437e+09	1.753874e+14	0.000000	999.975464	0.000000	201.699997	62.117252	-6.725497	NaN
1	1.325441e+09	1.753874e+14	2794.744466	4158.785290	0.026042	203.224997	62.099120	-6.710331	NaN
2	1.325444e+09	1.753874e+14	5589.488932	7317.595116	0.052083	204.749997	62.080987	-6.695165	NaN
3	1.325448e+09	1.753874e+14	8384.233398	10476.404942	0.078125	206.274997	62.062855	-6.679999	NaN
4	1.325452e+09	1.753874e+14	11178.977865	13635.214767	0.104167	207.799997	62.044722	-6.664834	NaN

Fig. 5. Snippet of the Maritime Mobile Service Identity track dataset combined with Kristina's trawler dataset used to train the model

These two files are joined. The timestamp column is converted to readable format. It is clear to that the timestamps collected are not in uniform interval.



	timestamp	mmsi	distance_from_shore	distance_from_port	speed	course	lat	lon	is_fishing
0	2012-01-01 17:00:00	1.753874e+14	0.000000	999.975464	0.000000	201.699997	62.117252	-6.725497	NaN
1	2012-01-01 18:00:00	1.753874e+14	2794.744466	4158.785290	0.026042	203.224997	62.099120	-6.710331	NaN
2	2012-01-01 19:00:00	1.753874e+14	5589.488932	7317.595116	0.052083	204.749997	62.080987	-6.695165	NaN
3	2012-01-01 20:00:00	1.753874e+14	8384.233398	10476.404942	0.078125	206.274997	62.062855	-6.679999	NaN
4	2012-01-01 21:00:00	1.753874e+14	11178.977865	13635.214767	0.104167	207.799997	62.044722	-6.664834	NaN

Fig. 6. Snippet of the dataset with readable timestamp feature used to trainthe model

For visualization correlation matrix to detect high correlated features, count plot to see the amount of data

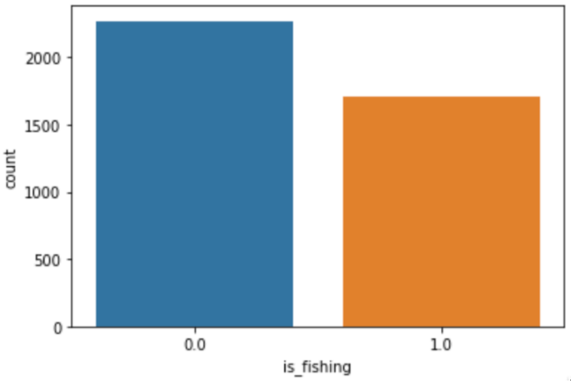


Fig. 7. Representing the count of the target or dependent variable

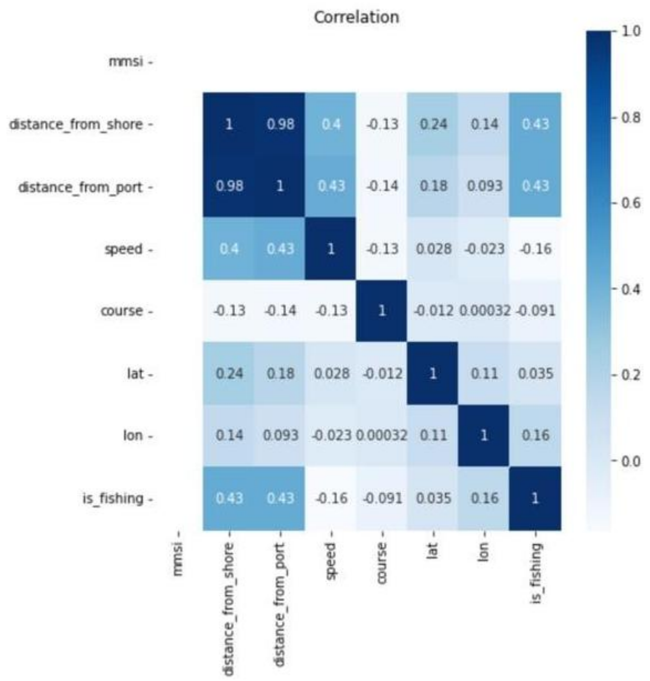


Fig. 8. Representing the corelation of all features with each other

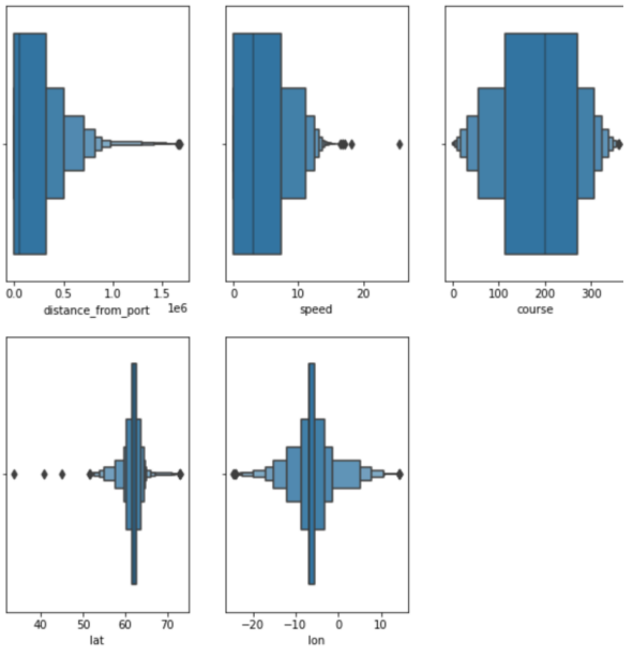
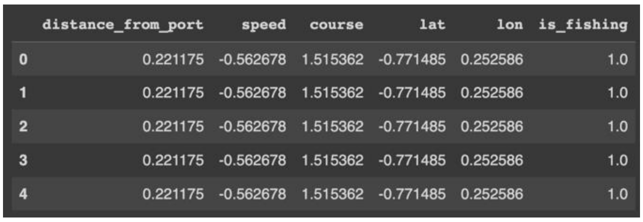


Fig. 9. Representing the boxplot of all features to determine any outliers in the dataset

indicating whether vessel was shipping or not, Box plot for distance from port, speed, course, latitude, and longitude to check for any noise / outliers in the dataset were plotted.

On plotting correlation matrix, it is seen that distance from shore and distance from port has high correlation. The machine learning models like Linear / Logistic regressions assumes that the input variable has little or no correlation i.e., no multicollinearity. Thus, these columns are removed. All unlabeled entities or missing data like latitude is dropped. As the timestamps collected are not in equal interval, it needs to convert the data into uniform intervals. The timestamp is converted from object to datetime object so it can be set as the index of the data frame to perform resampling on intervals of 1 hour. (As Object must have a datetime-like index to perform resampling using Pandas library). The entire dataset is resampled with intervals of an hour. The resampling function of Pandas inbuilt library is primarily used for time series data. A Time series is a sequence taken at successive equally spaced points in time. It is a Convenience method for frequency conversion and resampling of time series. After resampling is done timestamp column is dropped as it is redundant data for the model to train for our problem statement. Interpolation is performed to fill the missing data caused due to resampling.

Dataset is scaled using standard scalar. StandardScaler removes the mean and scales each feature/variable to unit variance. On testing it was found that scaling improved the performance of the model.



	distance_from_port	speed	course	lat	lon	is_fishing
0	0.221175	-0.562678	1.515362	-0.771485	0.252586	1.0
1	0.221175	-0.562678	1.515362	-0.771485	0.252586	1.0
2	0.221175	-0.562678	1.515362	-0.771485	0.252586	1.0
3	0.221175	-0.562678	1.515362	-0.771485	0.252586	1.0
4	0.221175	-0.562678	1.515362	-0.771485	0.252586	1.0

Fig. 10. Cleaned final dataset used to train model

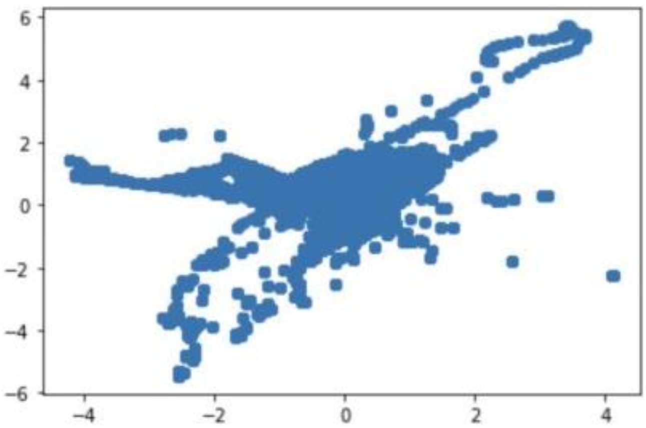


Fig. 11. Scatter plot of the latitude and longitude on the vessel with MMSI 175387414441613

A. Model Building:

We used multiple algorithms which are meant to solve classification tasks and compared the most optimal of those algorithms using Cross-validation techniques. The algorithms purposed in this paper are:

It is binary classifier meaning it can either classify an instance based on its features either as 1 or 0, TRUE or FLASE, YES or NO, etc. This algorithm is best suited for dataset which is linear separable (can be separated into 2 classes by using a S-shaped curve). It also performs probabilistic predictions and contains a Sigmoidal activation function, this ensures that he predictions(probabilistic) to not go beyond 1 and below 0, this uses a threshold value of 0.5 to classify a given instance as either 0 or 1. This algorithm is also best suited for low-dimensional data. Based on its assumptions logistic regression gave us an accuracy of 77.85% the reason behind such a dip in accuracy was because the data was not linearly separable, though it satisfied conditions like a linear relationship between the features(predictors) and target, more number of observations than the features, and no multicollinearity (redundancy was removed during data pre-processing/cleaning.

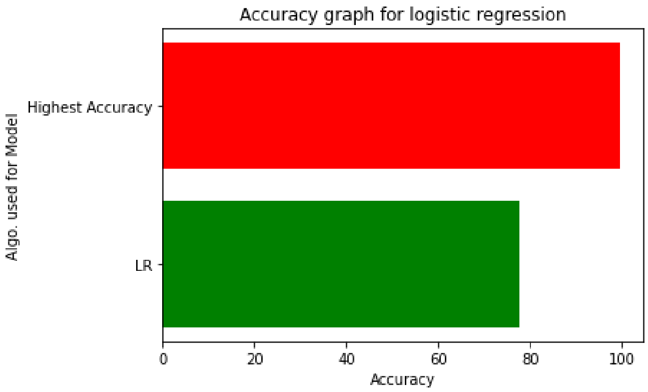


Fig. 12. Comparing model accuracy when using Logistic Regression

They provide us with non-probabilistic predictions, so they assign a data point to a class with 100% certainty. The main objective of SVM is to create a hyperplane/decision boundary in an n-dimensional dataset. The datapoints which are near to the hyperplane are called support vectors and they influence the behavior of the hyperplane. The distance from the SVM's classification boundary to the nearest data point is known as the margin, the goal of SVM is to maximize the margin.

Based on its assumptions we achieved an accuracy of 79.95%, the predictive accuracy is low because there was no clear margin of separation between classes and the number of dimensions were greater than the number of samples.

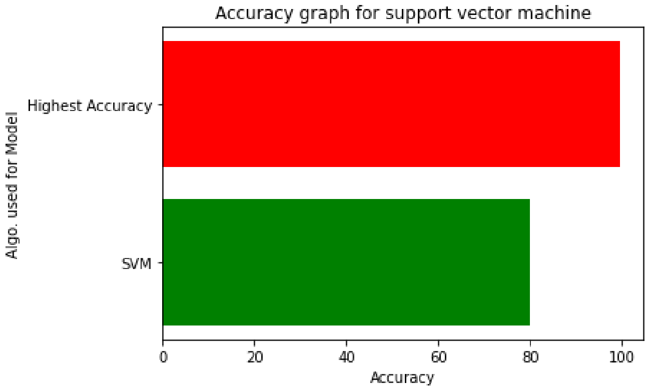


Fig. 13. Comparing model accuracy when using Support Vector Machine

It predicts a class label for a data instance directly and it can be modified to produce a probability-like score. It is a non- parametric algorithm i.e., it does not make any assumptions on the underlying data pattern. This algorithm does not go through a training phase it simply stores the dataset, and at the time when a new data instance occurs it assumes that similar things exist in proximity.

We achieved an accuracy of 95.69% which is severely overfitting the dataset, the reason behind this is an incorrect estimation for the value of k, a low value will overfit, and a high value will underfit the dataset.

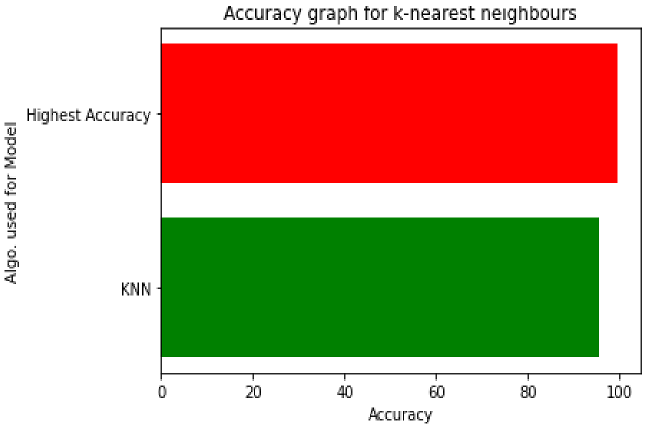


Fig. 14. Comparing model accuracy when using K-Nearest Neighbor

It is based on the principles of Bayes theorem which follows conditional probabilities, it is used to make probabilistic predictions. We achieved an accuracy of 78.37%, the assumptions made by the Naïve Bayes like: - no multicollinearity/independence between features and dataset is mutually exclusive (either 0 or 1 not both) completely satisfied the type of dataset that we used.

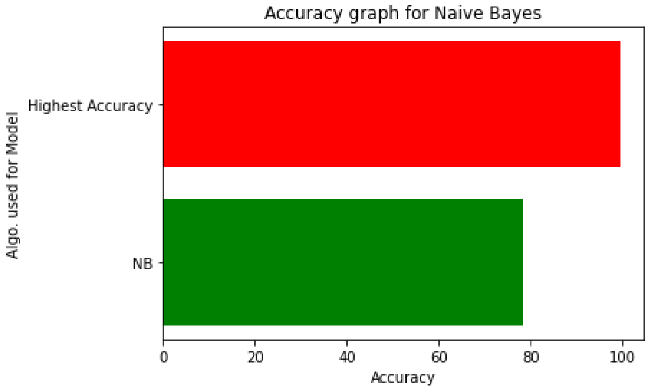


Fig. 15. Comparing model accuracy when using Naive Bayes classifier

A decision tree is an algorithm for supervised learning. It uses a tree structure, in which there are two types of nodes: decision node and leaf node. A decision node splits the data into two branches by asking a Boolean (0 or 1) question on a feature. A leaf node represents a class. We achieved an accuracy of 99.50% which is completely overfitting the dataset, the assumptions that the features need to categorical in nature and if the values are continuous, they are discretized prior to model building.

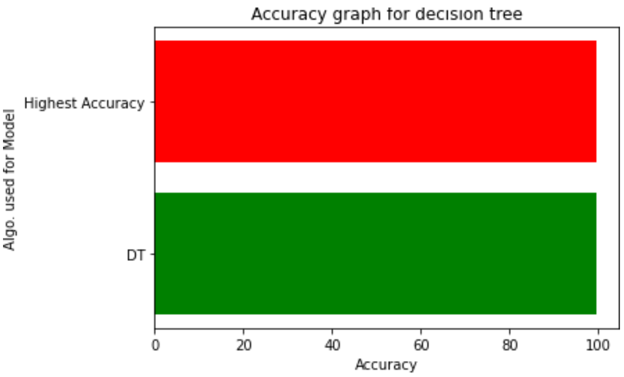


Fig. 16. Comparing model accuracy when using Decision Tree

It is an ensemble machine learning technique which is used to model complex relationships. It uses bagging to combine multiple predictive accuracies for various base estimators, Decision tree is used as a base learner. We achieved an accuracy of 99.73% which is completely overfitting the dataset because of the same assumptions made by a decision tree. RFC is always prone to overfitting which can be mitigated to some degree with pruning.

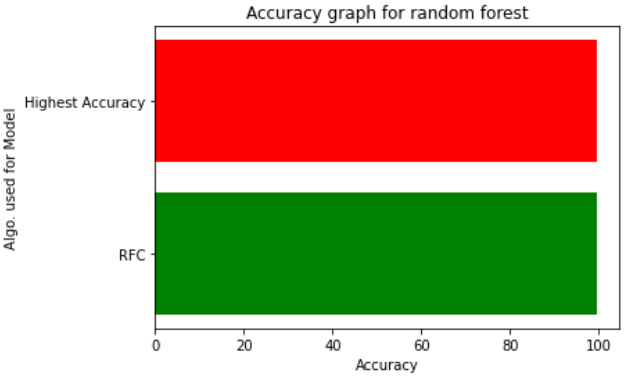


Fig. 17. Comparing model accuracy when using Random Forest

B. Cross Validation Techniques:

In each set (fold) training and the test would be performed precisely once during this entire process. It helps us to avoid overfitting. To achieve this K-Fold Cross Validation, we must split the data set into three sets, Training, Testing, and Validation, with the challenge of the volume of the data.

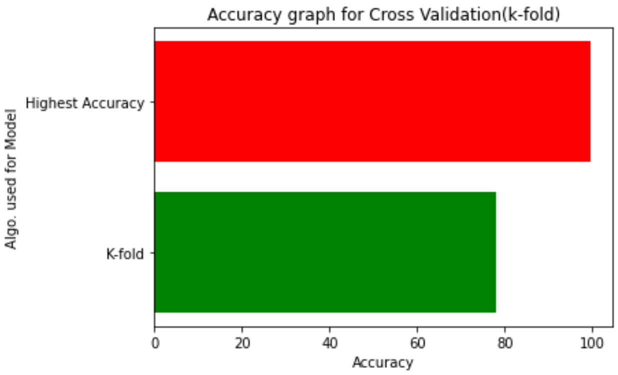


Fig. 18. Comparing model accuracy when using K-Fold Cross Validaton

It is used to find the optimal hyperparameters of a model which results in the most 'accurate' predictions. So, in contrast, this method not only provides us with the most optimal algorithm best also gives us hyperparameter estimates associated with the algorithm.

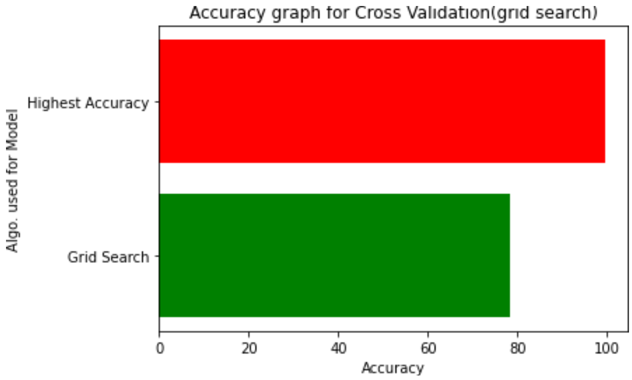
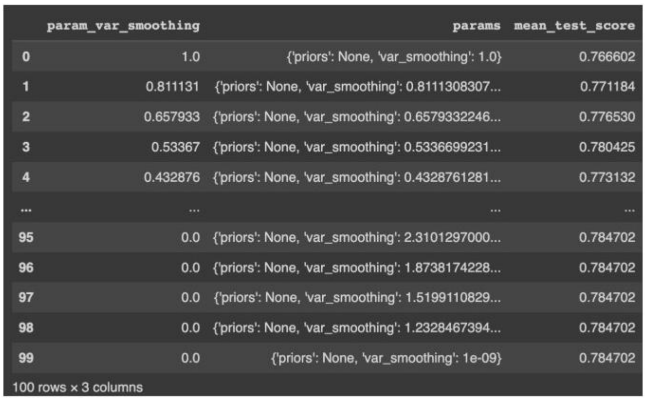


Fig. 19. Comparing model accuracy when using Cross Validaton using Grid Search



```
    param_var_smoothing                                          params  mean_test_score
0                   1.0           {'priors': None, 'var_smoothing': 1.0}        0.766602
1              0.811131  {'priors': None, 'var_smoothing': 0.8111308307...        0.771184
2              0.657933  {'priors': None, 'var_smoothing': 0.6579332246...        0.776530
3               0.53367  {'priors': None, 'var_smoothing': 0.5336699231...        0.780425
4              0.432876  {'priors': None, 'var_smoothing': 0.4328761281...        0.773132
...                 ...                                              ...             ...
95                  0.0  {'priors': None, 'var_smoothing': 2.3101297000...        0.784702
96                  0.0  {'priors': None, 'var_smoothing': 1.8738174228...        0.784702
97                  0.0  {'priors': None, 'var_smoothing': 1.5199110829...        0.784702
98                  0.0  {'priors': None, 'var_smoothing': 1.2328467394...        0.784702
99                  0.0         {'priors': None, 'var_smoothing': 1e-09}        0.784702
100 rows × 3 columns
```

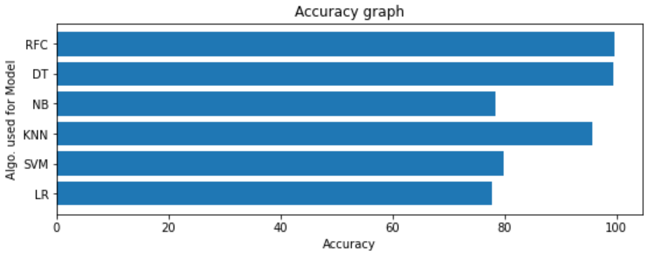
Fig. 20. Representing test score for 100 different parameters

V. CONCLUSION

Machine Learning models using Logistic Regression, SVM, KNN, Naive Bayes, DT and RFC were implemented. It was observed that the predictive accuracy for some models were rather high than others, specifically for Decision tree and Random Forest classifier the reason behind this was that the dataset was not very high dimensional (multiple features) and because of this reason the variance increased (Algorithms like DT and RFC can make very complicated assumptions which are not preferable for the dataset used) thereby causing the model to overfit the data. The phenomenon of overfitting can be reduced by: -

- Using less complicated algorithms or algorithms with fewer hyperparameters.
- Removing noise/outliers from the training data.
- Gathering more training data.

Lastly, Cross validation methods like k-fold and Grid-search were implemented. Parameters of Naive Bayes like priors, and var_smoothing along with the values were chosen as the most optimal hyperparameters used for learning. The best accuracy that was achieved using both the methods were 78.20 % and 78.23 % respectively.



Algorithm Name	**Accuracy achieved**
Logistic Regression	77.85
Support Vector Machine	79.94
K-Nearest Neighbors	95.69
Naïve Bayes	78.37
Decision Tree	99.49
Random Forest	99.72

Fig. 21. Comparing model performace and accuracy for predicting the vessel was shipping or not

VI. FUTURE WORK

Currently the model will only predict the outcome based on the data frames given in the csv file which are of a single ship. In the near future we would like to increase the number for large sum of ships. The project could be further improved by combining data sets from multiple fishing zones with expert-labelled data. The strength of the project is increased by integrating features that distinguish between maritime zonal restrictions based on international agreements and fisheries catching limitations based on regional legislation. This helps to more correctly separate the illegal and unreported fishing. A user-interface in form of an application/website can be built which will drastically improve the user-experience in understanding illegal fishing in a particular region. The user will also be able to compare the magnitude of illegal fishing of different ships.

ACKNOWLEDGMENT

We would like to extend our sincere gratitude to our faculty and mentor Professor Abhay Kohle, HOD B.Tech (Integrated)- Computer Engineering (NMIMS), for his advice and assistance in helping us to produce this paper. The opportunity to have been his students is something we value highly. We have benefited greatly from his expertise as a teacher and scholar. We hope to continue doing research with him.

REFERENCES

[1] V.K.G Kalaiselvi, et al. “Illegal Fishing Detection Using Neural Network” IEEE, 12 May 2022, ieeexplore.ieee.org/document/9767876.

[2] B.Padmaja, et al. “Catching Illegal Fishing Using Random Forest and Linear Regression Models” IJIREEICE, 6 June 2022, ijireeice.com/papers/catching-illegal-fishing-using-random-forest-and-linear-regression-models.

[3] B. Jyothi, et al. “Catching of Illegal Fishing with Data Analytics” Journalstd, 2021, journalstd.com/gallery/27-aug2021.pdf.

[4] Tamboli, Saeed. “Detecting Illegal Fishing Using AI.” Medium, 20 Oct. 2022, medium.com/@saeed_tamboli/detecting-illegal-fishing-using-ai-af1268aac5cf.

[5] Kroodsma, David. “Transshipment Data and Report - Global Fishing Watch.” Global Fishing Watch, 23 Feb. 2017, globalfishingwatch.org/data/transshipment-data-and-report.

[6] “Detecting Illegal Fishing with Machine Learning.” YouTube, 27 July 2022, www.youtube.com/watch?v=yDvM37OtKfo.

[7] https://www.vesselfinder.com/

[8] https://globalfishingwatch.org/data/reading-tracks-on-the-water/