Master Thesis

Exploring Fishing Vessel Activity Classifications using Machine Learning

by
Mark Ros

Presented for the degree Master Business Information Technology
With specialization Data Science & Business
At the University of Twente

In collaboration with

And University of Twente supervisors:
Dr. D. (Doina) Bucur, EEMCS
Dr. A. (Abhishta) Abhishta

September 2022

Cite as: Ros, M.F. (2022). Exploring Fishing Vessel Activity Classifications using Machine Learning

Acknowledgments

I would like to thank Dr. Bucur for her patience and help during the process of writing my thesis and Dr. Abhishta for joining me as my second supervisor. In addition, I would like to thank the Founder & General Manager of Sustainovate, Mark Sytse Ybema, for providing me with an interesting project topic. Also, his expert knowledge has helped me get up to speed within the fishing domain. Furthermore, I want to thank my supervisor and colleagues at Macaw, Robert Bakker, Suryashree Aniyan, and Ferdinand van Oostrom, for supporting me and providing much-valued feedback. I am happy to say I learned and grew while working on this thesis, and I am proud to be able to provide you with this interesting read.

Abstract

Seafood is vital for feeding the world's population, but awareness is growing about the environmental impacts of fisheries. Sustainovate develops a fish mapping tool that aims at saving fuel and time and leaving unwanted fish untouched. it uses environmental-, catch and fisheries activity data for this purpose. This research explores vessel activity classification using machine learning that is applied to the vessel's GPS data. Global Fishing Watch (GFW), a Google initiative, has developed a machine-learning model using a Convolutional Neural Network that classifies the activity of fishing vessels at a specific location and time. This model lacks accuracy, which does harm to the trust in fishing companies striving to fish more responsibly. For example, it may appear that vessels are fishing in prohibited areas while only reducing speed on their way to port. The aim of this research is to investigate the usefulness of classical machine learning models to develop a similar model as developed by GFW that could provide similar or more accurate results that is computationally cheaper and better to explain to clients for the Dutch pelagic fishing vessels that Sustainovate serves.

Since industry data could not be provided in time, scientific research data provided through the International Council for the Exploration of the Sea (ICES) open source database were used instead. A total of 70 trips from 3 international surveys (HERAS, IBWSS and IESSNS) conducted in the North Sea and North Atlantic over the past 12 years (2008-2021) were analysed. GPS data were extracted from 1 nautical-mile-interval echosounder recordings along the trajectory of the research vessels. The final dataset for this study contained over 5 million GPS records. In addition to the standard features found in the dataset, customised features were developed for this study: the ship's speed at the location and change over time, the ship's bearing change and the distance to the nearest port. Using these features in a single-point dataset and an aggregated segmented dataset averaged over 30-minute periods enabled the development of several machine learning models. The models developed are a Random Forest Classifier (RFC), a Naïve Bayes model with and without under-sampling, and a Support Vector Machine (SVM) model with and without under-sampling. Azure Auto ML was also used to see if a large-scale comparison of models would lead to better performance.

For the single-point dataset, the RFC model achieved a performance of about 87% for precision and balanced accuracy and at best about 76% for recall and F1 score. The other three models performed very poorly with statistics below 5%. The segmented dataset performed much better. With performance values above 80% for all models. The RFC in particular performed well with all statistics around 90%. The best performing was a voting ensemble model consisting of 8 XGBoost, RFC, and LightBM classifiers found by Azure Auto ML with 95% precision and the other metrics in the low 90%s.

Although the performance between the GFW- and the here developed model cannot be directly compared due to the different datasets used, the Azure Auto ML model and the segmented RFC model show similar performance to the GFW model, that had a precision of 98%, a recall of 94%, and an F1 score and accuracy of 96%. This research implies that a generally well-performing classifier for fishing vessel activities can be built with relatively little effort. It can therefore be expected that a continuation of this work using high-resolution data from the sector itself would very likely lead to better validation of the model and more accurate results. It could therefore contribute to the fish mapping tool developed by Sustainovate. In all use cases it is crucial to be able to explain the outcomes very well, which can be done much better with classical machine learning models than a deep learning model used by GFW.

Contents

List of Figures

List of Tables

1. Introduction

When talking about the non-stop availability of seafood, an increase in concerns about the environment is noticeable and the level of sustainability in everyday life is becoming more important. This is a topic that both consumers and suppliers have to adapt to. New technologies could enable suppliers to make better-educated decisions that can help in becoming more sustainable and/or less polluting during operation. One part of the seafood supply chain that could benefit from these new technologies given the biological and political changes is the fisheries. Therefore, this research will take a closer look into how a somewhat new technology, machine learning, could provide added benefit within this specific industry. Specifically, the possibility of classifying fishing vessel activities using their AIS/GPS signal is explored.

1.1. Problem description

According to the United Nations Sustainable Development Goal 2, by 2030, no one in the world should go hungry. Small pelagic fish such as herring, anchovy, sardines, and mackerel play an important role in the fight against hunger. While a large part of the small pelagic catches is usually destined to be processed into fishmeal to feed farmed fish or animals, the catches of the Dutch PFA fleet are only intended for direct human consumption. Approximately 90% of its fish are sent to low-income markets such as Ghana, Egypt, and Nigeria. Retention of the relative quota level of this fleet would ensure food supply to these regions.

Sustainovate AS is a Norwegian consultancy company with a focus on sustainable marine business development and innovation (Sustainovate, n.d. b). They are developing the OceanBox Smart Data Platform, a sensor-cloud software platform for the autonomous collection, analytics, and reporting of commercial fisheries sensor data to achieve data-driven responsible fisheries, with a focus on data-poor areas such as the Indian Ocean, South Pacific, and African coastlines (OceanBox, n.d.).

As part of their data services, Sustainovate is developing a fish mapping model that is to serve as a basic component for fish biomass estimations, required for sustainable fish management. Here, fish biomass means the total volume of fish, usually described in kg, present in a specific area. This fish mapping model aims to use a diverse selection of publicly available data and data generated by fishing vessels to make predictions on where the fish is located and the biomass size of fish. A key input source for this fish mapping model is catch location data. However, in many countries and fisheries catch data is not always (made) available.

In this lack of availability of catch location data lies the source of the problem this research project will try to address. The OceanBox service makes use of echo sounders which are onboard fishing vessels to scan the oceans and detect fish. This data could then be used as catch location data. This data is limited by the number and location of the fishing vessels, and many fisheries are not willing to share their catch data, resulting in the need for a new way of gathering catch location data.

One proposed method to achieve this is to use publicly available Global Position System (GPS)-data and/or Automatic Identification System (AIS)-data to predict fishing vessel activities as a proxy for catch data. Since AIS-transceivers are mostly mandatory for fishing vessels (Global Fishing Watch, n.d. e), it should become possible to ‘generate’ catch location data for any location that is being fished at.

This technique is already being used by the Global Fishing Watch (GFW), which is an initiative that aims to provide insights into commercial fishing activities, and to monitor and analyze human activity at sea (Global Fishing Watch, n.d. a; Global Fishing Watch, n.d. b). GFW uses a convolutional neural

network to classify fishing vessel activities. The image below (figure 1) shows such AIS Data with assumed catch locations, as interpreted by the globalfishingwatch.org algorithm.

Figure 1: AIS classification results for 'fishing' (https://globalfishingwatch.org/map)

The orange lines represent the travel path of the vessels while the white dots represent spots where the algorithm classifies the vessel to be fishing. While on first look the figure looks like it displays exactly what you would want, upon closer inspection it does have some major flaws. Looking at figure 2 below, a few incorrectly classified fishing locations have been marked. The area where these white dots have been placed is a 'no fishing' zone, meaning that it would be illegal to fish here, and therefore it is highly unlikely that these dots are correctly classified.

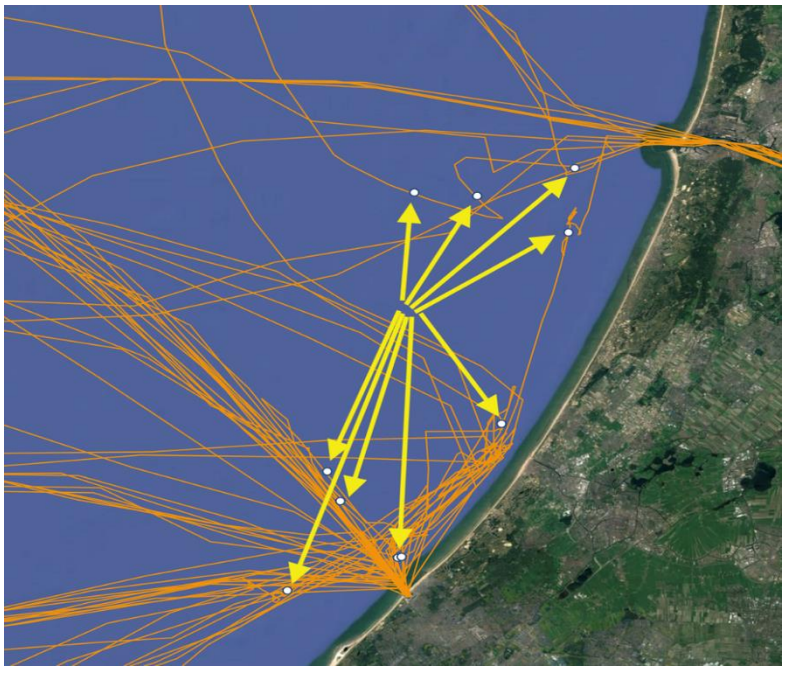


Figure 2: Wrong classification within 'No fishing'-zone (https://globalfishingwatch.org/map)

To reliably make use of the fishing vessel activity prediction as a proxy for catch location, the performance of this predictive model should be improved with a focus on reducing the number of false positives. However, besides wanting fewer false positives it is also crucial to be able to explain why a certain classification happens to convince potential client of the reliability of the service. Since GFW uses a convolutional neural network, which is a black box model, it is unclear why a data entry gets classified a certain way.

1.2. Research questions

To achieve the aimed contributions and research goals the following research questions are defined:

1. How can individual GPS/AIS-datapoints be pre-processed to be used for predictive modeling of activity classification?
   1.1. What techniques can be used to group individual GPS/AIS-datapoints into classified events?
   1.2. How can GPS/AIS-based events of different lengths be used together in machine learning models?
2. To what extent can supervised classification algorithms help in detecting fishing vessel activities, specifically fishing and non-fishing, based on GPS/AIS data?
   2.1. What features can be identified to help predict fishing vessel activity?
   2.2. What is the performance difference between various supervised classification algorithms in predicting fishing vessel activities?
3. How can the final machine learning model results contribute to the OceanBox fish mapping forecasting service?

1.3. Thesis outline

The rest of the thesis is structured as follows. Chapter 2 with provide information for the main topics that are relevant to the context of the thesis. Chapter 3 describes the literature that has been gathered to provide in-depth knowledge in the current state of the art. Chapter 4 compares some data science methodologies before explaining the approach taken during this research. Chapter 5 showcases the results obtained from applying the approach. Chapter 6 explains the conclusions that can be drawn from these results. Chapter 7 discusses the limitations of this research, explains the theoretical and managerial implications, and looks at possibilities for future research. The final section contains the appendix.

2. Background

In the problem description, an overall idea of the problem this research aims to solve has been described. This chapter will expand upon the context in which this problem lies.

2.1. Sustainovate's OceanBox

The OceanBox is an all-encompassing system for fisheries sensor data that starts with data generation and ends with extensive data analytics and data science. OceanBox has a goal to enable and support companies in optimizing their fishing strategies, avoid unwanted catch, and contribute to fish stock estimations (Sustainovate, n.d. b). The OceanBox consists of three key services, of which the first service also serves as the basis for the other services. This service is called the 'Essentials' and consists of data collection, data management, and reporting. (Sustainovate, n.d. b)

The data collection is done by the fishing vessels themselves (Sustainovate, n.d. b). Fishing vessels that want to contribute to the data collection will be equipped with sensors that generate data automatically. This data can be as simple as information about the location and speed of the vessel to more complex echosounder data that scans the ocean beneath the vessel. This echosounder data is preprocessed onboard the vessel to limit the size of the data and extract only the crucial information. Vessels then have the option to send this data to their cloud environment in Microsoft Azure. They then have the choice to share and join this data with other participants and other data systems such as fuel consumption data. This means that the data will always stay in the hands of the industry and clients have autonomy over their data. The data will be joined in a data lake which will store, manage, and process the data to be used for data visualization, analytics, and reporting. These visuals and reports are then shared with the industry which can help them in developing their fishing strategies. (Sustainovate, n.d. b)

Another service of the OceanBox that is being worked on at the moment is the Fish Biomass Estimation (Sustainovate, n.d. b). A lot of interesting information is being gathered by the equipped fishing vessel, combine that with publicly available ocean data and you can generate new insights by using machine learning methods. One of the insights generated is the presence and/or absence of fish biomass in the oceans. This service is currently still under development, but a lot of progress has already been made.

The final service OceanBox currently has to offer/planned is a fish species identification service, which will work in tandem with the above-mentioned Fish Biomass Estimation to also be able to identify the specific fish species present in a location.

2.2. Global Fishing Watch

Global Fishing Watch (GFW) is working on creating more transparency on the activities done by humans at sea. They do so "by creating and publicly sharing map visualizations, data and analysis tools" (Global Fishing Watch, n.d. a). GFW is a collaboration between Oceana, SkyTruth, and Google. Together, they have the tools and knowledge to generate and store larger amounts of data about the oceans. They focus on three different topics, Transshipment, Marine Protected Areas, and Commercial Fishing. (Global Fishing Watch, n.d. a)

Transshipment is the transfer of any catch of a fishing vessel to refrigerated cargo vessels called carriers at sea. This is done to eliminate the trip fishers have to take from and back to port, which is very time-consuming, while also increasing the freshness and thus the value of the catch (Global Fishing Watch, n.d. d). Since these activities take place out at sea, a lack of transparency exists due to limitations in monitoring. Therefore, any illegal activities, especially the trafficking of forbidden goods, could take place during these transactions without anyone knowing. GFW aims to bring more

transparency by providing clear information about these transactions, with the goal to limit the number of illegal activities that take place. (Global Fishing Watch, n.d. d)

Marine Protected Areas are specific areas set up in the oceans where the biodiversity has to be protected from any harm that can be done from outside of this area. This is crucial for said biodiversity as well as any people who are reliant on these areas to provide for themselves. Currently, only 3 percent of the oceans have been safeguarded, even though the United Nation was aiming for 10 percent by 2020. One of the main issues in setting up and achieving these Marine Protected Areas is the limited availability of data in these areas. GFW aims to provide more insights into the oceans to be able to better design, manage and monitor the Marine Protected Areas. (Global Fishing Watch, n.d. c)

Finally, Commercial Fishing focusses on the "for-profit practice of catching fish or other marine life by commercial fishing boats" (Global Fishing Watch, n.d. f). This activity plays a crucial role in the world economy and world food supply. However, overfishing and illegal fishing are underlying problems in the commercial fishing domain. These problems threaten the availability of fish and marine life in the long term. GFW is working on providing actionable information which enables proper governance of the oceans. They use satellite technology and machine learning to visualize where, when, and how vessels are fishing. One of their key goals is to provide this information for free to the public to help both fisheries and governing agencies to optimize their business practices. (Global Fishing Watch, n.d. f)

2.3. Effects of overfishing and illegal fishing

As mentioned, overfishing and illegal fishing are both major issues within the commercial fishing domain. The effects of these problems cannot be understated. When looking at the effects of overfishing a single type of fish, such as cod, can have this will become clear. Overfishing cod or other large predators influence the availability of those types of fish, possibly leading to the extinction of a species. However, those fish are part of an ecosystem and thus a food chain. Fewer numbers of these predators will lead to higher numbers of smaller fish, crabs, and shrimp (Scheffer, Carpenter, & de Young, 2005). This in turn will lead to smaller numbers of large-bodied zooplankton, leading to an increase in phytoplankton, which finally reduces the amount of nitrate (Scheffer, Carpenter, & de Young, 2005). Nitrate is a crucial nutrient that enables all oceanic plant life to grow (Widdicombe et al, 2009). So, the biological effect of overfishing stretches far further than just the specific fish that is being overfished.

Besides the biological effect overfishing has, another drastic effect can be noted when looking at illegal fishing. Illegal fishing is mostly done to ignore the restrictions put on specific species or ignore quotes set that are there to protect marine life and limit the chance of overfishing (Okey et al., 2007). By illegally fishing and thus ignoring these restrictions, the effects of overfishing as mentioned above also apply to illegal fishing. However, ignoring the ban on specific species, such as sharks and whales, is increasing the chances of these species going instinct even more. These bans are mostly put in place to protect already endangered species, so illegally fishing these species has a massive impact on their stocks (Okey et al., 2007).

Furthermore, illegal fishing can also have an impact on humankind. Illegal fishing has a detrimental effect on the economy of the fisheries that are fishing in the same or close-by areas (Okey et al., 2007). Profits of fisheries within the Gulf have been reduced by around 10% due to illegal fishing (Okey et al., 2007).

Besides, the financial impact on those within the vicinity of illegal fishing activities, the impact of illegal fishing also bleeds its way into the survivability of local communities. Along the African coasts many communities are dependent on the fish they catch to eat. The local fisheries here are too small to compete with the massive fleets of foreign fisheries. But not only does the fact that they are fishing close to these areas cause problems but just the presence of these far larger vessels also causes issues. Collisions happen, killing local fishermen. (Cannon, 2020)

By using data presented by GFW, interesting discoveries have been made. These discoveries include finding that large industrial-scale fishing was done 93% of the time in prohibited zones (Cannon, 2020). This indicates that there is a large-scale issue with illegal fishing and overfishing. It also shows that having this information available can enable governments and other agencies to act on these problems.

2.4. Concluding thoughts

During this chapter and chapter 1 it has become clear that two parties have played a large role in determining the direction of this research. First, there is the consultancy company with a focus on sustainable marine business development and innovation, Sustainovate. Their development of the fishery sensor data environment, OceanBox, is ongoing and will feature solutions to predict fish biomass presence and fish species identification. These services are developed to battle the negative effect of overfishing and illegal fishing, which can have drastic effects on nature, financial stability, and the survivability of local communities. To have these services make correct predictions, many data sources are needed. This triggered the interest in using fishing vessel classifications as a proxy to predict fish presence for their services. The second party is the Global Fishing Watch (GFW), which is an initiative to provide insights in commercial fishing activities. They have developed a service that classifies the fishing activities of fishing vessel. They use a black box model, a convolutional neural network, meaning that explaining how the model decides on a certain classification is very difficult. Therefore, this research will look for methods not used by GFW to achieve the same goal or perhaps even improve on it, while having a model that is explainable. To start, a deep dive in the academic literature has been done to determine a suitable competitor. The next chapter will discuss the findings.

3. Literature review

This chapter aims to provide insights into different machine learning techniques and algorithms, AIS/GPS data, and performance metrics. Paragraph 3.1 will describe the method used for literature collection for the algorithms and AIS/GPS-data knowledge. Paragraph 3.2 will provide an overview of the AIS/GPS data and paragraph 3.3 will do the same for the machine learning techniques and algorithms. Lastly, some performance metrics and over-and under-sampling techniques are discussed.

3.1. Review approach

This paragraph will detail the approach taken to gather the necessary information to develop an understanding of the different topics related to this thesis research. Different keywords were used to gather information regarding AIS/GPS data and machine learning models.

The databases used for the review are Scopus, accessed through the University of Twente LISA system, IEEE, and Science Direct. In case an article was not accessible, other sources such as Google Scholar were used to try to gain access. Due to the specificity of the subject, multiple searches were performed to gather a broader range of information. Search terms used were a combination of "machine learning" with combinations of, "GPS"; "Time series"; "activity"; "trip purpose"; "AIS"; "fishing". The searches were limited to the English language, the final publication stage, and article or conference paper document types. After reading the abstracts of the results, further selection was performed to remove any articles that did not prove useful to the topic at hand. Based on any useful results, a more in-depth dive into an interesting topic is performed as well.

Across all three sources, a total of 388 articles were found. After removing a lot of duplicates and removing articles based on the abstracts, the final selection consisted of 62 articles, of which not all were used in the end. During the development of this paper other searches were performed in case information was missing or a more in-depth understanding of a topic was needed. These articles have been selected based on convenience.

3.2. AIS/GPS-data

The first research question with sub-questions is heavily focused on pre-processing of AIS/GPS data. During this research, the main topic is AIS/GPS data of fishing vessels. Therefore, research on how this type of data can be used is crucial. The number of articles found regarding both AIS/GPS data and fishing vessels was very limited. However, when the components of the question are dissected, it becomes clear that the specific context of fishing vessels is not all that important. Preparing GPS data to be used in a predictive model can be used in various contexts such as predicting animal behavior (Browning et al., 2018), detecting travel modes (Xaio, Cheng, & Zhang, 2019), detection of traffic characteristics such as street crossings or traffic lights (Munoz-Organero, Ruiz-Blaquez, & Sánchez-Fernández, 2018), earthquake detection (Gitis, Derendyaev, & Petrov, 2021), and of course vessel behavior classification (Kontopoulos, Chatzikokolakis, Tserpes, & Zissis, 2020). Although these contexts are rather different from each other, similarities in the data processing approach can be seen in all of them. Two trains of thought can be applied here, a single-point analysis or a segmented analysis.

3.2.1. Single-point analysis

The single-point analysis is the most straightforward and easiest to implement. Here the specific moment in time a certain state occurs is not important, only the characteristics of that state matter. Using the context of this research as an example, a ship will have a certain speed and change in bearing at any point in time. Using speed and bearing as features might enable the possibility of

classifying an activity at a certain point in time. Therefore, at $t = x$ the features are $speed = 7km/s$ and $change\ in\ bearing = 45°$ and at this time the vessel is fishing. On the other hand, if this vessel is piloting from point A to point B the features might be $speed = 18km/s$ and $change\ in\ bearing = 2°$. A clear distinction can be made here between the two scenarios where it becomes clear that a vessel that is fishing is steering a lot more and going a lot slower than when the vessel is piloting from A to B. While this provides a very high level of detail in which classification takes place, it also requires very distinct differences between the features of the classes. If the vessel has a data point in which they were to go in more of a straight line during fishing, the features might look like $speed = 7km/s$ and $change\ in\ bearing = 5°$. Based on speed alone you might be able to say, "This vessel is fishing", but based on the change in bearing alone you might say "This vessel is piloting from A to B". This can prove to be challenging while optimizing the performance of the Machine Learning model. (Qin, 2020)

3.2.2. Segmented analysis

To combat some of the issues sketched in the previous subsection a solution could be to use segments of data. Here, the time aspect which was mostly irrelevant for single-point analysis suddenly becomes important. A segmented analysis builds upon the basis set by the single-point analysis. Features are engineered, such as speed and change of bearing, but instead of looking at the state of these features at a specific point in time, the focus shifts to the state of these features during a specific period. This is something that is used often in travel mode research (Xiao, Cheng, & Zhang, 2019; Soares, Revoredo, Baião, de MS Quintella, & Campos, 2019). In travel mode research, the researchers are trying to predict what form of transportation (i.e., bus, car, bike, walking) is used during a trip. A trip can contain multiple forms of transportation, someone walks 5 minutes to the bus stop, rides the bus for 30 minutes, and then walks 10 more minutes to the appointment the person must be at. During this trip, two forms of transportation were used during three segments of the trip. This shows that segments are parts where a single form of transportation is used. Using this for the current research is not as far-fetched. Each segment is then a specific activity a vessel is performing; first a vessel is piloting from shore to a fishing spot (segment 1), then the vessel is fishing (segment 2), then the vessel will return to shore (segment 3). By looking at segments instead of single points it becomes clear that a small period of similar behavior between the classes (i.e., change of bearing is the same between fishing and piloting as shown in the previous paragraph) will not lead to the same problems. (Xiao, Cheng, & Zhang, 2019; Soares et al., 2019; Zhang, Dalyot, Eggert, & Sester, 2011)

To go from a dataset suitable for single-point analysis to segmented analysis, some work is required. The features used in the single-point analysis must be aggregated in the segments. The change in bearing at $t = x$ must become the average change in bearing or frequency of change in bearing in $segment = s$. However, it is necessary to have a clear condition as to where a segment stops and starts. The literature provides a method of determining segments. This method is to determine a specific time window, such as a minute. All data gathered within said time window would then be used to detect activity change points. Such a change point would happen when a clear distinction can be seen between the previous time window and the following time window. This would then divide the route into segments by classifying all windows before the change point as a single class and all windows after the change points as another class. (Xiao, Cheng, & Zhang, 2019; Soares, Revoredo, Baião, de MS Quintella, & Campos, 2019; Zhang, Dalyot, Eggert, & Sester, 2011)

3.2.3. Fourier Analysis

From a segmented analysis, a different approach can be taken by looking into the Fourier Analysis. This type of analysis is a segmented analysis at its base since specific periods of time will be analyzed, however, the approach here will be a bit more complex than mentioned above. Like the segmented analyses, a Fourier analysis is only possible for time series data.

A time series dataset consists of variables gathered at a specific interval to be able to predict a new variable $y()$. An example of this is the heartbeat that is visible on an electrocardiogram (ECG). A time series is closely related to a signal. In a time series, the $y()$ variable is dependent on a time value, whereas a signal is not dependent on time but can be dependent on a different type of variable.

A time-based signal can exist in two forms, continuous and discrete. When thinking of an ECG again, the continuous time series will show that at every possible moment in time a specific level of an electrical charge of the heart can be measured. So, at $t = 1\ s$, at $t = 2\ s$, but also at $t = 1.278\ s$. This signal is, therefore, continuous in nature. To use the information stored in this continuous signal, it needs to be converted into a discrete signal. To do this, the rate at which data is gathered is needed. For example, the electrocardiograph might be set up at 50Hz, meaning that per second, 50 measures are taken. This results in time being available at the points $t = 0.02$, $t = 0.04$, but not at $t = 0.024$, indicating a discrete dataset. This frequency, or rate, at which the data gets gathered, or sampled, is called the sampling rate. This sampling rate is very important since it can drastically alter the accuracy of the representation of the underlying naturally continuous signal. This is visualized very well in figure 3 below. (Ataspinar, 2018; Ahmed & Nandi, 2019)

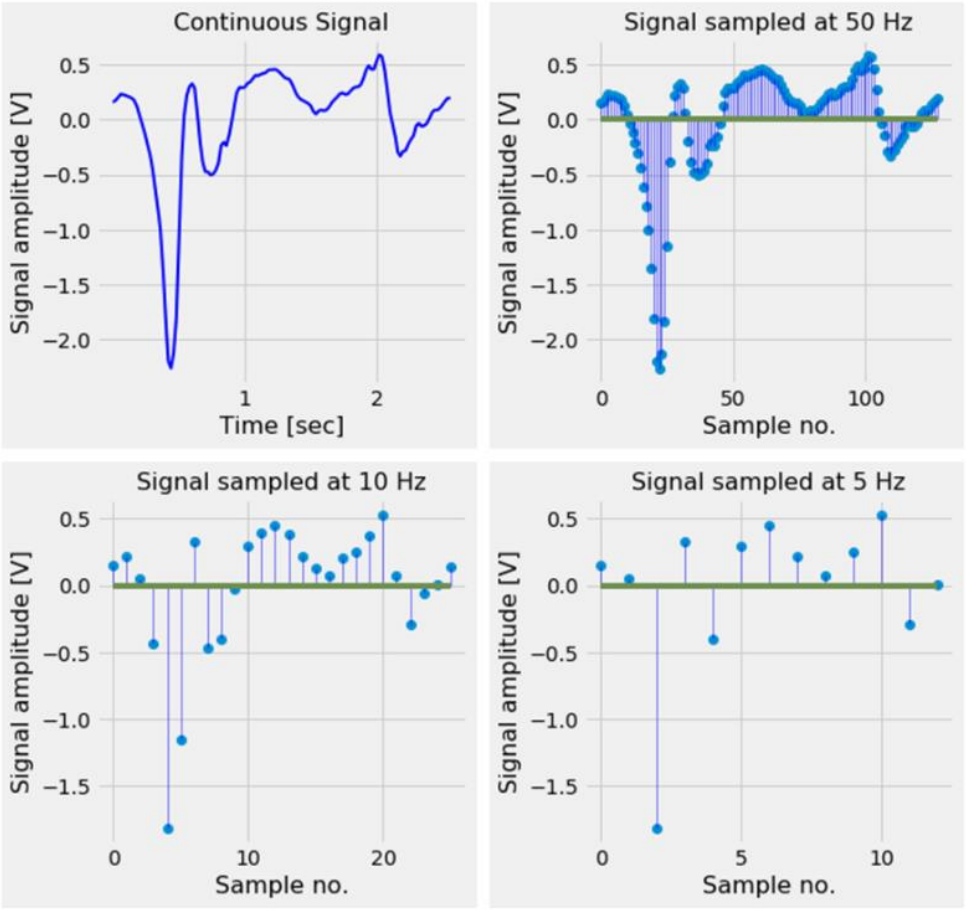


Figure 3: A continuous signal sampled at different sampling rates (Ataspinar, 2018)

Determining an appropriate sampling rate can be done by making use of the Nyquist rate, which represents double the highest frequency present in the sample. The sampling rate should then not be any smaller than the Nyquist rate to obtain a proper discrete signal. (Ataspinar, 2018; Ahmed & Nandi, 2019)

A signal is easily interpretable by humans, but monitoring a signal constantly is not feasible. The heart rate of a normal heart of a person in rest should be very consistent. However, patients just recovering from cardiac surgery might have a heart problem called Atrial Fibrillation (AF) which causes the heart to work irregularly for periods of at least 30 seconds. This is very time-consuming to detect manually, however, when transforming this to a frequency domain it can become very clear that one 30-second period has a much higher frequency than a regular 30-second period. (Ataspinar, 2018; Ahmed & Nandi, 2019)

The transformation of a discrete signal to a frequency signal is done by Fourier Transformation. Fourier Transformation has two components, the period, and the wavelength. The period represents the time between repetitions of a periodic signal while the wavelength represents the duration of a periodic signal. The period can be inversed to get the frequency. This frequency can then be used to plot a frequency-amplitude graph showing the characteristics of a signal. This graph can contain multiple peaks, indicating the presence of multiple underlying signals which would not be detectable in the time domain. Thinking of the ECG example again, a normal heart rate would have a peak at 1 Hz on the graph given a resting heart of 60 beats per minute. If AF occurs, the heart rate might contain a small period of 90 beats per minute and a small period of 110 beats per minute, and then a small period of 60 beats per minute. Then, three different frequencies can be counted, one at 1.50 Hz, one at, 1.83 Hz, and one at 1 Hz. This has a greatly different characteristic than just a consistent 1 Hz, indicating the presence of AF. This indicates the useability of time series and Fourier Transformation to detect patterns in time series data. (Ataspinar, 2018; Ahmed & Nandi, 2019)

Fourier Transformation has added benefit in the context of this project as well. By transforming a segment using the Fourier Transformation a sort of ‘fingerprint’ of that segment can be created. When a vessel is fishing, the fingerprint will display the frequencies at a different place than when a vessel is going from A to B, therefore, indicating two different fingerprints. This fingerprint could be compared to a fingerprint that represents a specific vessel activity to be classified.

3.2.4. Conclusion

The three approaches to handling AIS/GPS data mentioned above could be applied to this research. Due to the effort-to-reward ratio of the single-point and segmented analysis, these will be investigated during this research. These methods have been proven in the literature to work for similar applications and are thus reliable to use for this research. The Fourier Transformation shows a lot of potential and might outperform the other two methods. However, due to the time investment needed to correctly implement this method and the lack of literature backing this method for GPS research, it will not be used during this research.

3.3. Machine learning techniques/algorithms

After developing a method of handling the AIS/GPS data, this data needs to be put into a specific model to be trained and tested. During the literature review, a couple of techniques/algorithms kept showing up. These models were a Hidden Markov Model, Support Vector Machines, Random Forest Classifier, and a Naïve Bayes. Furthermore, the Global Fishing Watch specifically uses a Neural Network algorithm (Kroodsma et al., 2018), and during the development of this research, Auto ML of the azure environment was used which resulted in a Voting Ensemble model as the best performer. Therefore, these techniques/algorithms will be discussed in the following subsections.

3.3.1. (Convolutional) Neural Networks

As mentioned, the Global Fishing Watch uses a Neural Network for its product. Specifically, they use a Convolutional Neural Network (Kroodsma et al., 2018). Such a Neural Network has been developed to replicate the way a human brain works (Wang, 2003; O'Shea & Nash, 2015). This is done by having multiple layers of neurons interconnected which will lead from input data to output data. The first layer is the input layer, then there is a single or multiple hidden layers before finishing in the output layer. The neurons between layers are connected and these connections have weights. The weight determines the level of change that can occur to the values of the neuron between layers. The higher the weight, the bigger the possible change, and thus the more important the connection between the nodes is. Figure 4 shows a visual representation of this architecture. (Wang, 2003; O'Shea & Nash, 2015)

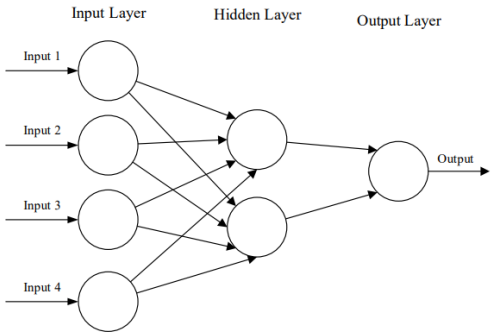


Figure 4: Architecture of a Neural Network (O'Shea & Nash, 2015, p2)

A Neural Network is mostly used in a supervised way but can be used unsupervised (Dike, Zhou, Deveerasetty, & Wu, 2018). Specifically, a Convolutional Neural Network is used mostly for pattern recognition within images (O'Shea & Nash, 2015). This is because images contain a lot of parameters that a traditional Neural Network has difficulties dealing with. A Convolutional Neural Network does not have this problem since this type of network makes fewer connections between neurons than the traditional Neural Network. The traditional Neural Network creates a connection between each Neuron, as shown in figure 8. A Convolution Neural Network connects the neurons by groups, as shown in figure 5.

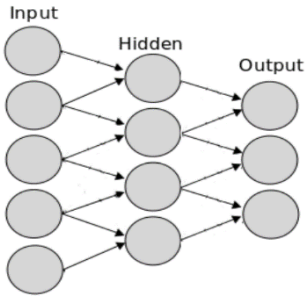


Figure 5: Architecture of a convolutional Neural Network (Stojov, Koteli, Lameski., & Zdravevski, 2018, p3)

3.3.2. Support Vector Machines

Support Vector Machines (SVM) are a great way of assigning labels to an object. The way this is done by plotting the data points and then trying to separate the different labels or classes by using a hyperplane (Suthaharan, 2016; Noble, 2006). This is visualized in figure 6.

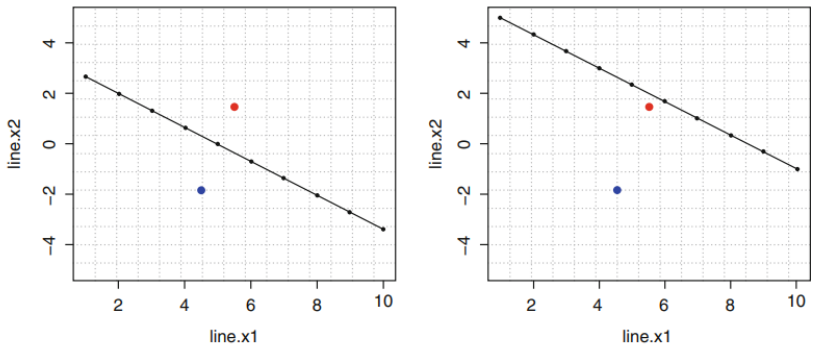


Figure 6: Seperating hyperplane of an SVM (Suthaharan, 2016, p211)

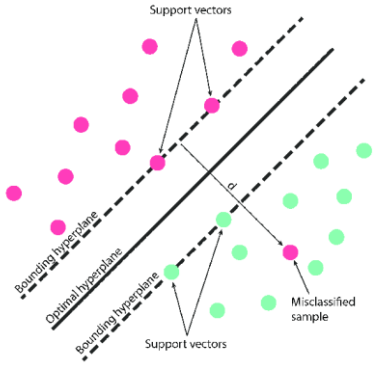


Figure 7: Bounding and optimal hyperplane of a SVM (Cardoso-Fernandes, Teodoro, Lima, & Roda-Robles, 2020; Suthaharan, 2016)

Here, two classes can be seen represented by the red dot and the blue dot. The goal of an SVM is to separate these classes by plotting a line between the two dots. As humans, we can clearly see that the line on the left does this perfectly while the line on the right does not. The SVM will thus need to calculate what line is optimal and it does this by calculating a line that is roughly in the middle of the data points. To get to this line, the SVM calculates the distance a line must travel to reach the closest datapoint under a perpendicular angle for both data points. This is done by using two hyperplanes and then selecting the middle between these planes that indicates the optimal hyperplane. This can be seen in figure 7. (Cardoso-Fernandes, Teodoro, Lima, & Roda-Robles, 2020; Suthaharan, 2016)

The SVM explained above uses the linear kernel. While this kernel is the most straightforward and easiest to understand, it is not the only kernel. The kernel is relevant for transforming the data from one dimension to the next dimension. This is one of the strengths of an SVM since the dimension in which the data exists does not determine the effectiveness of the separating hyperplane. As can be seen in figure 8, the yellow and purple dots cannot be separated using a linear line, when looking at the left figure. However, by lifting the data into a different dimension a clear distinction can be made and a linear plane can be placed between the yellow and purple dots. Herein lies one of the strengths of SVM. To transform the data into a new dimension, a so-called kernel function has to be chosen. This kernel function is responsible for the way the dimension gets transformed. It has been established that the data cannot be separated using a linear line, thus by transforming the data using a polynomial kernel a circular separation line can be drawn. This can be visualized by transforming the data using this kernel. After doing so, the data can be linearly separated in the new dimension. (Suthaharan, 2016; Sicotte, 2018; Nanda, Seminar, Nandika, & Maddu, 2018)

The most used kernel functions are linear, polynomial, radial basis, and sigmoid (Nanda, Seminar, Nandika, & Maddu, 2018).

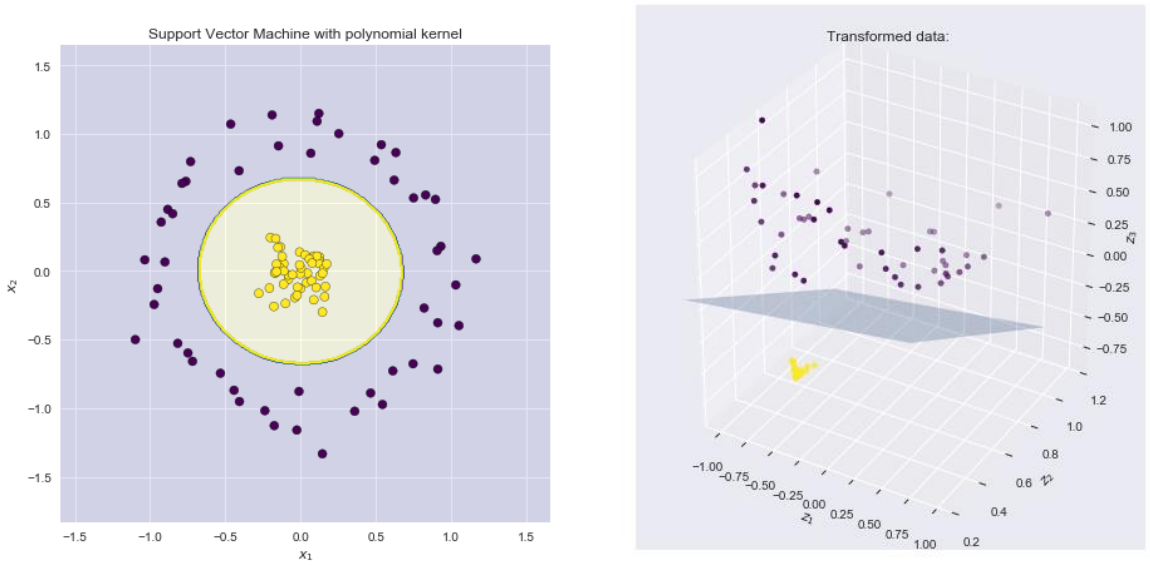


Figure 8: SVM dimensionality change (Sicotte, 2018)

One of the issues of an SVM, however, is the reduced performance when using an imbalanced dataset (Cardoso-Fernandes, Teodoro, Lima, & Roda-Robles, 2020).

3.3.3. Random Forest Classifier

The Random Forest Classifier (RFC) is a grouped classifier since it consists of multiple tree classifiers. Each tree decided on what the class of a specific input should be and by combining this a final classification can be made by the RFC. The way a single tree determines this class is by using nodes and paths, where each node represents a test, and the outcome of that test determines the following path to take. The more important a test is, the higher up in the tree the test is done. After doing each test, a decision can be made on what class the input is. Within an RFC, each tree has different tests and different paths that lead to a classification, resulting in an array of classifications. By taking the average prediction the RFC then makes the final classification. An RFC is very useful since it can take any form of data, numerical and categorical, it is not affected by multicollinearity, it is easy to interpret, and delivers good results. However, it does tend to be easily overfitted. (Myles, Feudale, Liu, Woody, & Brown, 2004; Song & Ying, 2015)

3.3.4. Naïve Bayes

The Naïve Bayes method is a very basic method. It is rooted deeply in the basics of statistics and is, therefore, a probability classifier. It counts the frequencies and combinations of specific values in the data to calculate the probability of the input being a specific class. A drawback is that it assumes independency between variables which is mostly not the case in real-world scenarios, but it does provide quick calculations and the results are easily interpretable. (Saritas, & Yasar, 2019; Ren, Lee, Chen, Kao, Cheng, & Cheung, 2009)

3.3.5. Voting Ensemble

Voting Ensemble is a technique that combines multiple models that each cast a vote to then make a final classification. This is to a certain degree similar to a Random Forest Classifier. As mentioned, a Random Forest Classifier averages the predictions of multiple decision trees. The Voting Ensemble does something similar but is not limited to only looking at decision trees. Many different types of models can be applied that each cast a vote on what the classification should be, given the input. Just

like with the RFC, after each model has cast its vote, they get aggregated to then make the final classification. (Saha & Ekbal, 2013; Dietterich, 2000)

3.3.6. Conclusion

Concluding the research into the machine learning techniques/algorithms, not a single technique can be said to be the best performer. Each technique has its own strengths and weakness. A Convolutional Neural Network is the technique used by GFW. For this reason, the performance of this model will be used to compare to the newly developed models. The SVM is a technique that shows great promise given the ability to separate classes in multiple dimensions, but the reduced performance with imbalanced data will limit the effectiveness of this technique. The Random Forest Classifier is a method that has very good overall performance and is very explainable, while not being affected by multicollinearity. However, Random Forest Classifiers are known to be sensitive to overfitting. The Naïve Bayes method is a very fast method and has a strong and simple underlying logic. It is, however, a naïve method, meaning that it might not show the best results in a complex real-world scenario.

Given these strengths and weaknesses, the SVM, RFC, and Naïve Bayes methods will be used during model development. These three techniques have truly different styles of classifying, resulting in a clear range of different types of models that are investigated during this research.

Lastly, after performing an Azure Auto ML run, a Voting Ensemble method came out as the best performing model, resulting in it also being used for comparison.

3.4. Performance metrics

During this research, the different machine learning techniques are going to be compared with one another, as well as the model developed by GFW. To perform this comparison a selection of performance metrics has been made. The selected metrics are Precision, Recall, F1-Score, and Accuracy. Important here to note is that when using imbalanced data, the accuracy score that is used is a balanced accuracy. This selection has been made mostly since these metrics help in showcasing how well false positives are filtered out of the data but also because GFW only has these metrics available to compare with (Kroodsma et al., 2018).

The four performance metrics can be calculated using a confusion matrix. This is a table that visualizes and summarizes the predicted classes against the actual classes as shown in figure 9. Here, four values can be seen. These are the True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). These indicate what the values were and whether the predicted value was right or wrong. Using these values, the performance metrics can be calculated. (Singh, Elhoseny, Singh, & Elngar, 2021; Visa, Ramsay, Ralescu, & Van Der Knaap, 2011)

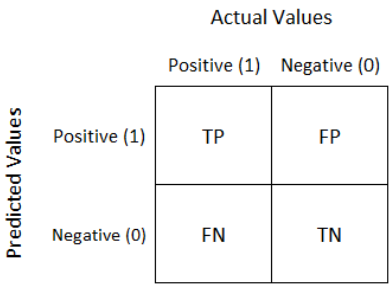


Figure 9: Confusion Matrix Example

The way to calculate these metrics is as follows:

$$Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN}$$

$$F1-score = \frac{2*Precision*Recall}{Precision+Recall} \qquad Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

$$Specificity = \frac{TN}{TN+FP} \qquad Balanced\ accuracy = \frac{Recall+Specificity}{2}$$

Here, the precision can be seen as the quality of the prediction of the positive class. If the precision is high, then the quality of predicting positive values is good. Recall then shows the quantity. The precision might be high, but if most of the actual positive values get classified as negatives then the overall performance of the prediction is still poor. Recall is the metric that showcases how many of the actual positives are being classified correctly. The F1-score represents the harmonic mean value between the precision and the recall. Specificity is the metric that showcases how many of the actual negatives are being classified correctly. The accuracy represents how many values across the entire set get classified correctly. Lastly, the balanced accuracy shows the overall accuracy of the correct classification, but by taking the change of data imbalances in account. (Visa, Ramsay, Ralescu, & Van Der Knaap, 2011)

3.5. Over- and under-sampling

One of the issues with trying to create predictive models for real-world issues is the imbalance in data that can occur. This imbalance in data means that one class exists far more often in a dataset than the other. Within the context of this research that is the case as well. A fishing vessel is far more often not fishing than that is it fishing. This results in the dataset containing a multitude more non-fishing classes than fishing classes. This imbalance in data can result in biases or overfitting of the data, or it could result in harder to interpret the results. As mentioned in the previous chapter, the performance metrics that are going to be used are precision, recall, F1-score, and accuracy. By having imbalanced data, what can happen is that the precision, recall, and F1-score are drastically low, while

still having an accuracy of over 90%. This can happen when a dataset contains massively more classes of class 0 than of class 1. The model then classifies all instances as class 0, resulting in a precision, recall, and F1-score of 0% while the accuracy is still high since most of the data was classified correctly. A solution to this is over- and/or under-sampling. Oversampling is creating more data of the minority class to rebalance the data. This could be done by duplicating existing data points. This results in a more balanced dataset; however, a bias can occur. Under-sampling is the other method. Here, entries from the majority class are randomly selected and deleted until the dataset is balanced. A problem here is that it is impossible to know what information is being thrown out using this method, however, this method has proven to be rather successful. (Liu, 2004)

3.6. Concluding thoughts

The literature provides multiple ways to use AIS/GPS-data to be used in machine learning applications. The most classical and straightforward approach is to use the individual data entries as a single moment and thus to classify each of these moments. However, since AIS/GPS-data is a time series data source it is also possible to group the data into segments over a specified period. In doing so, it is possible to classify an entire segment. Furthermore, multiple machine learning techniques are suitable to be used for classification. During this research three techniques have been manually developed, which are a Support Vector Machine, a Random Forest Classifier, and a Naïve Bayes model. Besides those three, a Voting Ensemble model came out as strongest during an automated machine learning comparison using Azure Auto ML. To determine the performance of these models four metrices have been chosen including precision, recall, F1-score, and (balanced) accuracy. Finally, to be able to combat imbalanced data the techniques of over- and under-sampling can be applied to alter the balance of the data to be more balanced.

Using the information gathered from the literature, new models can be developed to match or improve on the model of GFW. The models found in the literature are classic white box machine learning models, ensuring the explainability of the models. The next chapter will compare four data science methodologies after which one methodology is used for developing the new models.

4. Research Methodology

In this chapter, four different data science methodologies will be discussed. After comparing these four, one of the methodologies has been chosen to work with and will be applied to the current project.

4.1. Data science methodology

Throughout the lifetime of data science, different methodologies have been developed to successfully complete data science projects. To aid in the development of this thesis project, a closer look into some of these methodologies will be taken. Based on prior knowledge and through Google Scholar using the term "Data Science methodologies" a few recurring well-documented methodologies were found. These five methodologies will be highlighted in the following paragraphs.

4.1.1. Knowledge Discovery in Databases (KDD) process

In 1996 Fayyad, Piatetsky-Shapiro, and Padhraic Smyth, came up with the Knowledge Discovery in Databases (KDD) process. This process focuses on using data mining techniques to "identify valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad et al., 1996, p. 30). The KDD process itself consists of five steps. Before and during these steps the application domain needs to be considered and understood. The five steps of the KDD process can be seen in figure 10 and can be described as follows (Fayyad et al., 1996):

1. Selection: During this step, a target dataset will be created, or the focus will be shifted to a subset of variables or data samples.
2. Pre-processing: By cleaning and/or pre-processing the data, a workable dataset will be generated.
3. Transformation: This step aims to reduce the number of unneeded variables in the dataset and to transform data to represent its correct features.
4. Data mining: The data mining part is a somewhat larger step. During this step, many decisions need to be made with regard to what the data mining should do, with what algorithm, and the actual application. The goal here is to recognize patterns in the data.
5. Interpretation/Evaluation: interpret the patterns discovered and evaluate whether this is the required knowledge or whether to return to any of the previous steps.

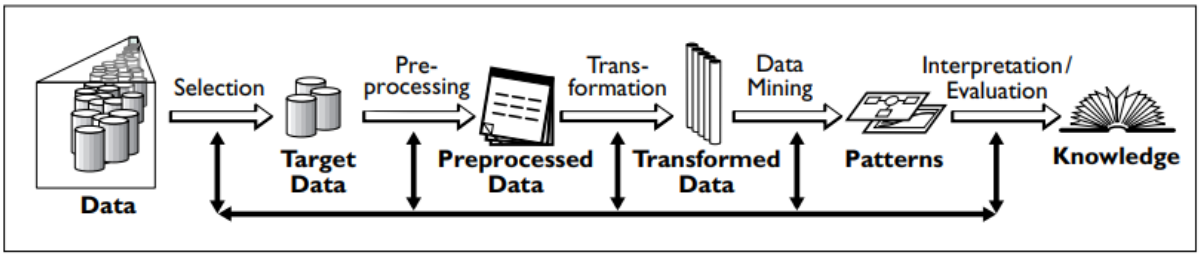


Figure 10: Overview of the KDD process steps (Fayyad et al., 1996, p 29)

4.1.2. CRISP-DM

While Fayyad, Piatetsky-Shapiro, and Padhraic Smyth were working on the KDD process, SPSS and Teradata were developing the CRISP-DM process (Wirth & Hipp, 2000; Foroughi & Luksch, 2018). This process is cyclical and consists of six steps which can be seen in figure 11 and can be described as follows (Wirth & Hipp, 2000; Foroughi & Luksch, 2018):

1. Business Understanding: The first step in the cycle is the business understanding in which the underlying problems and goals will be described. Alongside several requirements, a project approach within the data mining domain can be developed.
2. Data Understanding: During the data understanding step the required data will be collected and analyzed to get the first insight into the data and to be able to identify any problems with the data.
3. Data Preparation: The previous step delivers a single or multiple raw datasets. These datasets need to be formatted and cleaned in a way to be usable for further analysis.
4. Modeling: The modeling step consists of using the data and different modeling techniques/algorithms to reach the required output.
5. Evaluation: After creating the model, it is important to evaluate whether the model performs sufficiently to be able to achieve the goals set during the Business Understanding step.
6. Deployment: When the developed model or models are evaluated and accepted the code representation of said models can be deployed to function in the operational environment. This means that the model should be able to consistently provide relevant insights based on new input data.

During any of these steps, it is possible to revert to a previous step. In figure 11 the most common stepbacks are modeled. In 2014, CRISP-DM was voted the most used methodology for data mining or data science projects (Piatetsky, 2014).

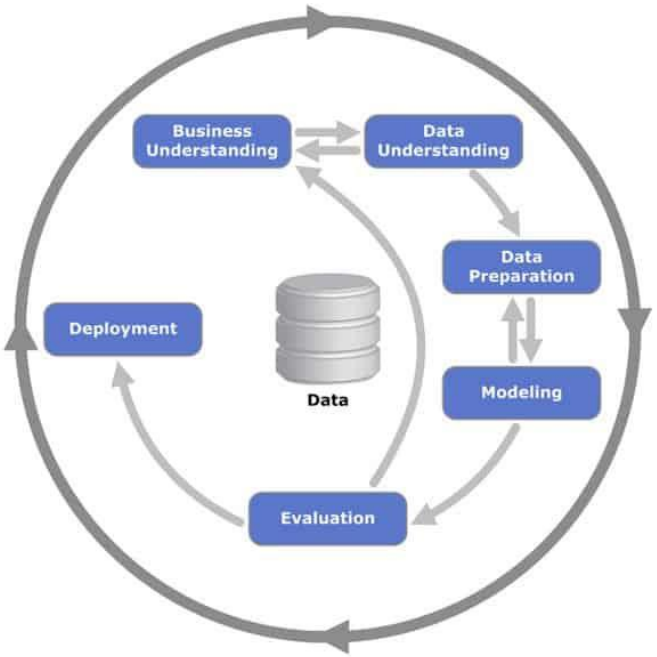


Figure 11: CRISP-DM diagram from Data Science Process Alliance (2021)

4.1.3. Foundational Methodology for Data Science (FMDS)

A more recently developed methodology is the Foundational Methodology for Data Science (FMDS) by IBM in 2015 (Rollins, 2015). This methodology is similar to the previous two, however, FMDS has altered its methodology in such a way to incorporate newer techniques such as big data, text analytics, and some form of process automation (Rollins, 2015). The methodology consists of 10 steps which can be seen in figure 12 and can be described as follows (Rollins, 2015):

1. Business Understanding: Just like the CRISP-DM first step, FMDS starts with creating a proper understanding of the underlying problems to set up proper project goals. Crucial here is to involve the sponsor/client of the project during the project. (Rollins, 2015)
2. Analytic Approach: This step focuses on the conceptual approach the data scientist will take to solve the problem concerning the statistical and machine learning methods they are going to use to reach the set goals. (Rollins, 2015)
3. Data Requirements: Based on the chosen approach the needed data will have to meet some requirements which will be determined in this step. (Rollins, 2015)
4. Data Collection: During this step, any useful data will be gathered. It could happen that during this step not all data requirements can be met through the available data. If this is the case, it is important to revise the requirements and/or to look for new or additional data sources. (Rollins, 2015)
5. Data Understanding: When enough data is gathered, preliminary data analysis can take place in which descriptive statistics and visualizations can be used to acquire initial insights about the data. This helps in assessing the data quality and understanding the data. (Rollins, 2015)
6. Data Preparation: With FMDS the data preparation part is expanded past the traditional description of data preparation. This step will therefore include traditional data cleaning and data formatting but will also contain concepts from newer techniques. Concepts such as feature engineering will belong in this step as well. During feature engineering, new variables/predictors/features are created based on analysis from unstructured or semi-structured data such as patient documentation. Since this step will usually take the most time in a data science project the addition of automation is most notable here. (Rollins, 2015)
7. Modeling: By using the analytic approach and the prepared dataset a predictive or descriptive model will be developed in which the dataset is used as a training set. This process is highly iterative due to performing adjustments based on the intermediate results. Trying multiple algorithms and changing their parameters is often done to find the best-performing model. (Rollins, 2015)
8. Evaluation: It is important to be sure that the quality and the efficacy of the model that has been created tackles the project goals sufficiently before deployment. This is done by calculating different diagnostics measures or performance metrics on a testing dataset that is separate from the training data. The testing dataset contains the features as well as the outcome, enabling the comparison of the model against reality. (Rollins, 2015)
9. Deployment: When the model passes the evaluation and the sponsor/client approves, it is deployed to the production environment. The deployment can be either done by embedding it into the architecture or by creating a report with recommendations. The extent of deployment is dependent on the level of evaluation. (Rollins, 2015)
10. Feedback: After the model has been deployed into its environment feedback can be gathered on its way of functioning. The feedback can then be used to further adjust the model until it performs to satisfaction. Gathering feedback is another part where automation can often be used. (Rollins, 2015)

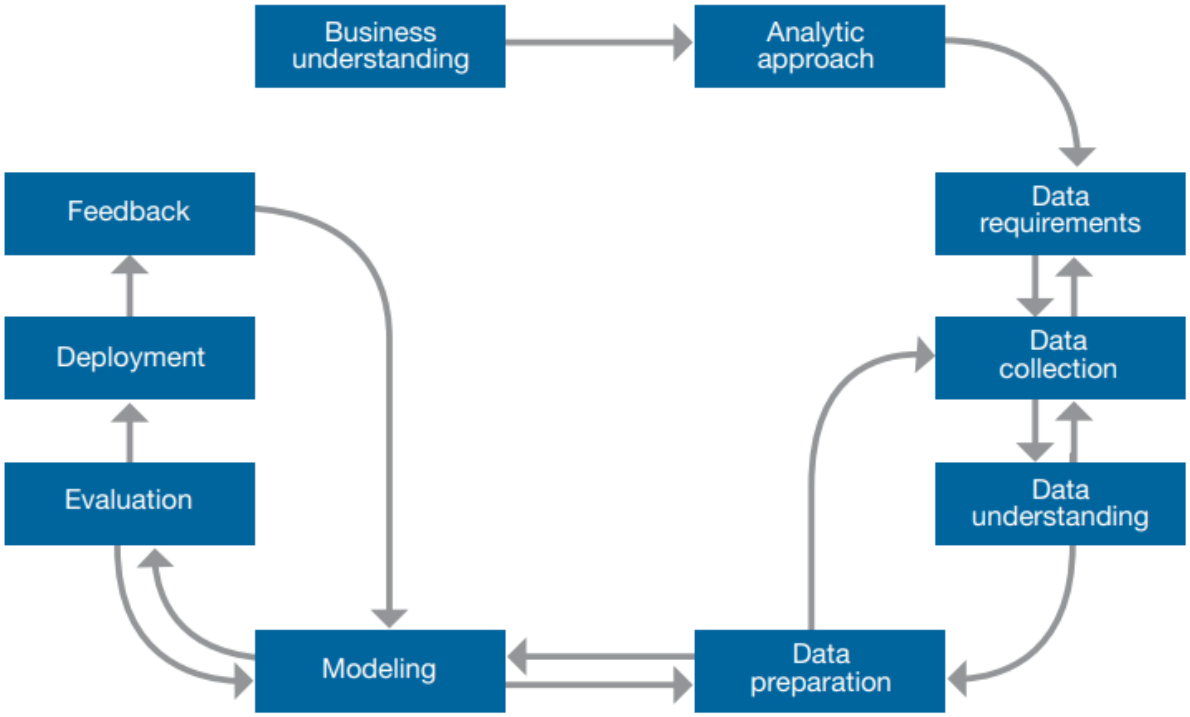


Figure 12: Foundational Methodology for Data Science (Rollins, 2015)

4.1.4. Team Data Science Process

Team Data Science Process (TDSP) is a methodology developed by Microsoft and has a larger focus on teamwork and agility within data science projects. The higher-level concepts of TDSP can be used in a project which employs a different project process such as CRIPS-DM or KDD. The TDSP process consists of 5 stages or phases. These phases can be seen in figure 13 and can be described as follows: (Marktab, Alexbuckgit, & V-kent, 2021a; Foroughi & Luksch, 2018)

1. Business Understanding: During this stage, the goal is to define what variables and metrics will serve as indicators to test whether the project is a success, so the project goals should be clear during this stage. Subsequently, what data sources should be acquired will also be identified during this stage. (Marktab, Alexbuckgit, & V-kent, 2021b)
2. Data Acquisition and Understanding: The data acquisition and understanding stage has the goal to acquire the required data sources and then explore this dataset to see any faults in the data and to check whether the data is useful at all. After this, the data can be cleaned and a pipeline can be set up to, if necessary, gather and clean fresh data automatically. (Markta, Alexbuckgit, & V-kent, 2021c)
3. Modeling: Feature engineering is the first step during this stage. A usable set of features will be selected and/or created which can then be used for model training. After selecting a suitable algorithm, the model can be trained and when it performs sufficiently, it can be altered to be deployed to production. (Marktab, EdPrice-MSFT, Alexbuckgit, & V-kent, 2021d)
4. Deployment: After the model is created it can be operationalized. The crucial part of this stage is the part where the data pipeline and the predictive model are working automatically and can be used by the end-users. (Marktab, Alexbuckgit, Igayhardt, & V-kent, 2021e)
5. Customer Acceptance: After the final product is finished, feedback should be gathered from the customer/end-user to see whether they accept the final product. (Marktab, Alexbuckgit, & V-kent, 2021f)

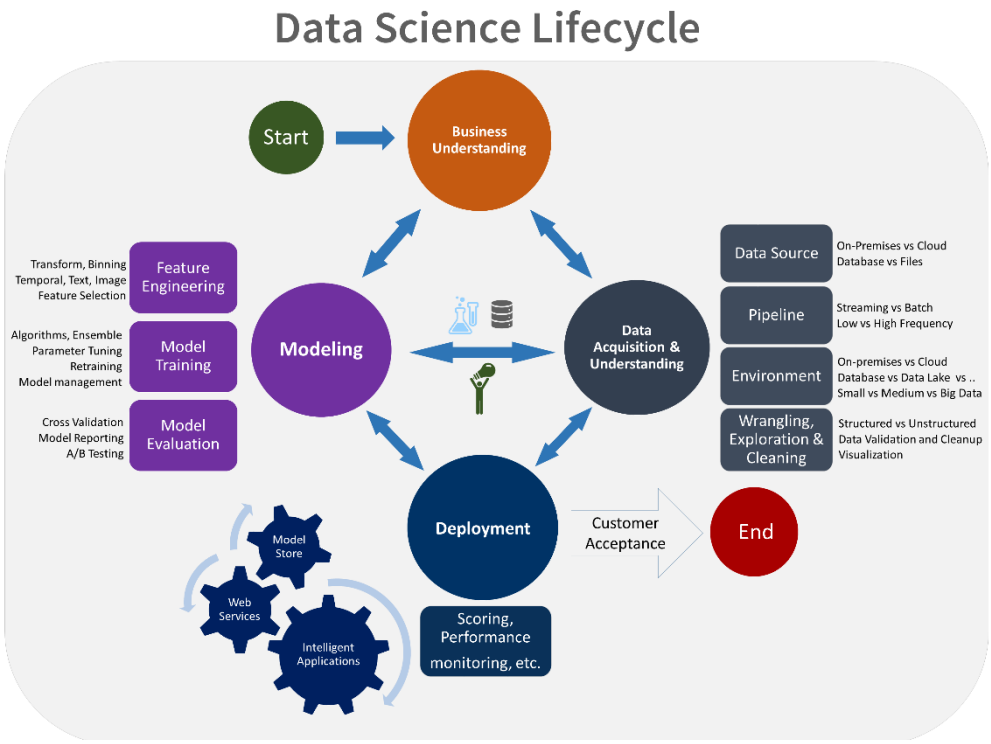


Figure 13: Team Data Science Process (Marktab, Alexbuckgit, & V-kent, 2021)

4.1.5. Conclusion

The four methodologies described have a very similar overall approach, however, some differences between the approaches exist, enabling the elimination of some methodologies.

The KDD methodology states that the business understanding should be kept in mind during the entire process, however a specific stage for this step is missing. This makes this methodology less suitable for projects in which the domain is unknown to the project members, as is the case in this project.

One of the main values of the TDSP methodology is the focus on teamwork. However, during this project most, if not all, of the work will have to be done by a single person and thus the main benefit of this approach does not apply to the current project.

The CRISP-DM and the FMDS methodology are very similar to each other, meaning that both methodologies would fit very well. Both methodologies feature an iterative and complete process from business understanding to deployment. The main difference is that within FMDS the step data preparation has been split up. Therefore, both methodologies would make a great fit.

The final choice fell on the FMDS methodology. The reasoning behind this decision is that while the CRISP-DM methodology is currently the most used Data Science Methodology on the market (Piatetsky, 2014), it is also a rather old methodology. While this does not provide any issues for this project, it can be debated on how future-proof this methodology is based on recent and emerging technologies and trends in the data science domain. The FMDS methodology explicitly states that new technologies such as big data and text mining can be easily applied using this methodology (Rollins, 2015). This project will contain the first version of a deliverable with a data pipeline and predictive model, which might get expanded greatly in the future. Therefore, using a methodology that will be more future-proof, might prove to be very beneficial when expanding the deliverable.

4.2. Business Understanding

At the start of the thesis period, conversations with the founder of Sustainovate have been held to come up with a suitable project. During these conversations, an overall idea of what Sustainovate does became clear and the project goals were developed and agreed upon. After determining the scope of the project it became clear what knowledge was to be gathered to accomplish the set goals. To do this, multiple papers and websites were read, and conversations were held to better understand the fishing domain, the OceanBox product, and the current state of similar models to the one that has been developed for this project. Some key contributors here are the website of the International Council for the Exploration of the Sea (ICES, https://www.ices.dk), the website of Sustainovate (https://sustainovate.com), the website of Global Fishing Watch (https://globalfishingwatch.org), the Owner of Sustainovate, an expert from Sustainovate who is part of developing the OceanBox, and a team lead of Global Fishing Watch. The output of this step is a clearly stated problem description with the aimed contribution of this project to said problem. This is done through specific research questions that are answered in this thesis report. The descriptions of these can be found in chapter 1. Subsequently, the context or background will be described through this step, which can be found in chapter 2.

4.3. Analytic Approach

To ensure that the conceptual approach is sound, an extensive literature review was done to figure out what machine learning algorithms/methods are most fitting for the given goals. During this literature review, different machine learning concepts have been highlighted as well as methods to handle GPS/AIS data.

The output of this step is a better understanding of the different machine learning algorithms/methods and GPS/AIS-data handling methods which results in a selection of different techniques to be used during data preparation and model development. The final chosen methods can be found in chapters 3.2.4. and 3.3.7.

As a method of determining the performance, the following performance metrics have been chosen, Precision; Recall; Accuracy; F1-Score. Of these metrics, the most important one is Precision, since one of the main goals of this project is to specifically limit the false positives compared to the performance of GFW.

4.4. Data Requirements

This research project is based on using specific datasets that are available publicly and acquired through partnered companies. Therefore, the data requirements are more like data scope. These datasets still need to have some basic attributes, however. The data that will be needed will consist of a GPS signal of specific ships paired to a timestamp. Also, catch data of these ships with an accompanying timestamp will be needed to be able to train the models.

4.5. Data Collection

After the theoretical work, the more practical side of the project is started. The first thing to do was to collect the data. For this project, a selection of research datasets has been used. This data has been downloaded from ICES at https://acoustic.ices.dk/submissions. On this website, different surveys from shipping vessels, such as acoustic trawlers, can be found. A single submission consists of two files. The first file contains the acoustic data gathered from the vessel itself. This data contains information about the vessel such as a timestamp, the vessel's location through latitude and longitude, some metadata parameters, and some other variables that are unimportant to this project.

The second file is a data file containing biotic data, meaning data portraying the biological information gathered from the fishing hauls. This consist of three different tables. First, the Haul table contains data regarding some details about the time and location (latitude and longitude) a haul has been initiated and how long this haul took. The second table has information about what has been caught in the previously mentioned haul, specifically, what type of species has been caught, and optionally biomass information. Finally, the optional biology table contains information about the individual fish found in a haul containing the weight of each fish for example.

ICES gathers information from a lot of different surveys. For this project, the scope was set to three of them, HERAS, IBWSS, and IESSNS. These abbreviations stand for North Sea Herring and Pelagic Ecosystem Survey, International Blue Whiting Spawning Stock Survey, and International Ecosystem Summer Survey, respectively. For these surveys, the vessels all have to fish for similar fish and in doing so have the same method of fishing. This means that the data available will have a uniform meaning. An overview of some information about these files can be found in table 1. (Bergès, Sakinan, Berg, Lusseau, Schaber, & O'Connell, 2021; Marine Institute et al., 2018; Nøttestad et al., 2021)

One crucial fact about these surveys is that their sampling rate of gathering the data is per nautical mile (ICES, 2015). This means that the frequency at which data is gathered is irregular when looking from a time perspective. When a vessel has a high speed, it will reach a nautical mile in less time than when it has a low speed. This greatly impacts information availability during low-speed segments.

During later development, a new dataset was introduced. This dataset contains information about all ports across the world. This information is acquired from the World Food Programme (WFP) which is the largest humanitarian organization (World Food Programme, z.d.).

Finally, some AIS data from a few ships on the coast of Oman will be used as a validating set once the models have been developed. The data has been provided by exactEarth. This data consists of only AIS data and does not have any catch data to verify. Table 1 shows some information about this dataset. The verification will be done by Sustainovate. This form of verification will, unfortunately, be a weak form of verification since Sustainovate will look at whether the results of the model "seem to match with what they know".

Dataset	Location	Time	Datapoints	Vessels	Time granularity
ICES HERAS	North Sea & North Atlantic Ocean near UK & Ireland	1 to 2 month periods between 2009 and 2021	202364 for Acoustic 1635 for Biotic	53	Entry every 11,5 minutes
ICES IBWSS	North Atlantic Ocean near UK & Ireland	1 to 2 month periods between 2011 and 2022	15884 for Acoustic 115 for Biotic	13	Entry every 11 minutes
ICES IESSNS	North Atlantic Ocean near Iceland	1 to 2 month periods starting July 2021	11967 for Acoustic 280 for Biotic	4	Entry every 12 minutes
Oman	Coast of Oman	1 year from May 2021 till June 2022	13281 for Acoustic No Biotic	1	Entry every 10 minutes
World ports	Worldwide	N.A.	3582	N.A.	N.A.

Table 1: Overview of all datasets

4.6. Data Understanding

After all data is gathered explorative data analysis is performed within a local environment. This is done to assess the data quality. To start, however, the data needs to be cleansed. Each CSV file contains all available tables on a single screen. This results in the inability to simply import the data using Python and start analyzing. Therefore, each CSV file must be opened and the rows containing the useless tables must be deleted.

The data that is left consists of 70 CSV files containing the acoustic data and 70 CSV files containing the matching and useful biotic data. One matching set is data from a single vessel, thus 70 vessels are in the complete dataset. In total, this results in 5581382 rows of data for the acoustic set and 2034 rows of data for the biotic set. The acoustic data consists of 29 columns and the biotic data consists of 45 columns. A detailed description of each column can be found in table 2 and table 3 on pages 25 and 26.

An example of what the data looks like can be found in table 16 (Appendix 1) and table 17 (Appendix 2). As can be seen, a lot of columns consist of 'NaN' values. In some cases, these columns are partially filled, however, this is such a small percentage that the columns cannot be used reliably. These statistics can be found in table 18 and table 19 (Appendix 3). So, the first step was to remove all (nearly) empty columns. This left the acoustic set with 24 columns, which can be found in table 20 (Appendix 4), and the biotic set with 13 columns, which can be found in table 21 (Appendix 5).

From these remaining columns, a decision needed to be made on which columns were going to be useful for the data preparation and the model. Looking back at the Data Requirements it becomes clear that the most basic information needed is location and time. Therefore, from the acoustic set the columns LogTime, LogLatitude, and LogLongitude are required to match these requirements. Furthermore, the column LogValidity is very crucial to make sure that the information that is used is also valid. The biotic dataset contains similar columns to be used. Here, the columns HaulStartTime, HaulStartLatitude, and HaulStartLongitude are needed to get the basic time and location information. HaulValidity will again serve as a check to ensure the usage of correct information. Lastly, the column HaulDuration is gathered to be able to extend the number of fishing data points, more on this later. To get an idea of what a fishing route looks like in the acoustic set, three examples of a route of three ships, one of each survey, can be seen in figure 14.

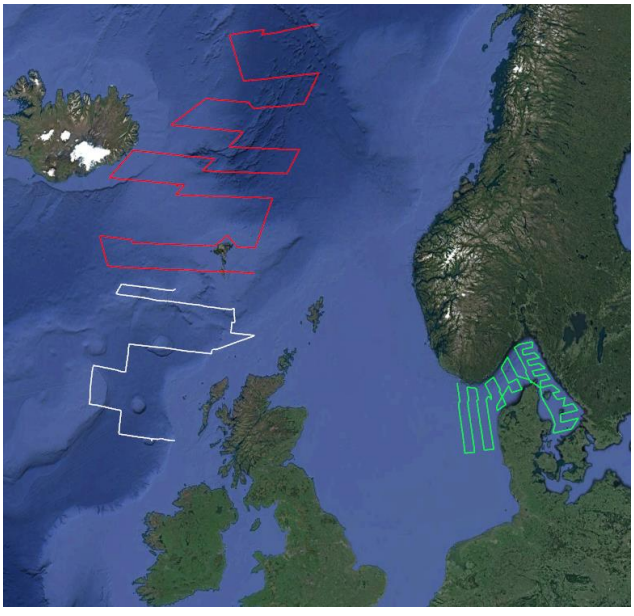


Figure 14: Example of routes of the ICES Data

For the Oman dataset, a single '.txt'-file is used. This file contains 16 columns of which only 7 are filled. These columns are the MMSI of the vessel, which is a unique identifier of the AIS signal, timestamp, latitude, longitude, Speed over Ground (SOG), Course over Ground (COG), and NavStatus. The NavStatus column is very interesting here. What that column shows is the navigational status of the vessel. This status can be any countable number between 0 and 15, where, for example, 0 means 'Under way using engine'. The interesting part is status 7 'Engaged in Fishing'. This shows us that the vessel is fishing, enabling the opportunity to properly test the performance of the model. The issue is, however, that when plotting this data it becomes clear that this status is not reliable for the goal of this project. Figure 15 shows that the vessel is fishing while being docked and close to the port, as is shown by the green dots, which is known to be not true. Furthermore, the vessel has only 893 of the 13485 rows as 'not-fishing' activity, which is known to not be too little. The reason for it being unreliable is most likely due to the fact that the status is manually set by the crew of the vessel (Marine Traffic, z.d.) and thus the fishing status is not turned off when the vessel finishes fishing.

Lastly, the world ports dataset consists of 25 columns, which can be found in table 22 (Appendix 6). Of these columns, the only information that is interesting for this project is the latitude and longitude columns indicating the location of a port. In total this dataset contains 3582 rows of data.

Figure 15: Oman dataset NavStatus fishing locations port area

Acoustic dataset:

	Data Type	Description
Data	string	Key field used to identify record type
Header/Record	string	Key field used to identify header and record rows
LogDistance	decimal	Log distance in nautical miles
LogTime	string(16)	Log time in ISO 8601 format: YYYY-MM-DDThh:mm[:ss] or YYYY-MM-DD hh:mm[:ss]
LogLatitude	decimal	Log latitude in decimal degrees
LogLongitude	decimal	Log longitude in decimal degrees
LogOrigin	string	Log origin - AC_LogOrigin, see Options
LogLatitude2	decimal	Log latitude2 in decimal degrees - used to specify log latitude end
LogLongitude2	decimal	Log longitude2 in decimal degrees - used to specify log longitude end
LogOrigin2	string	Log origin2 - AC_LogOrigin, see Options - used to specify log origin end
LogBottomDepth	decimal	Log mean bottom depth in metres
LogValidity	string	Log validity - AC_LogValidity, see Options
SampleChannelDepthUpper	decimal	Upper channel depth (m) relative to surface
SampleChannelDepthLower	decimal	Lower channel depth (m) relative to surface
SamplePingAxisInterval	decimal	Data ping axis interval value
SamplePingAxisIntervalType	string	Data ping axis interval type: Vocabulary - AC_DataPingAxisIntervalType, see Options
SamplePingAxisIntervalOrigin	string	Data ping axis interval origin: Vocabulary - AC_DataPingAxisIntervalOrigin, see Options
SamplePingAxisIntervalUnit	string	Data ping axis interval unit: Vocabulary - AC_DataPingAxisIntervalUnit, see Options
SampleSvThreshold	decimal	SvThreshold
InstrumentID	string	Reference to the InstrumentID in the Instrument record
CalibrationID	string	Reference to the CalibrationID in the Calibration record
DataAcquisitionID	string	Reference to the DataAcquisitionID in the DataAcquisition record
DataProcessingID	string	Reference to the DataProcessingID in the DataProcessing record
EchoTypeID	string	Reference to the EchoTypeID in the EchoType record: conditional mandatory with DataSaCategory
DataSaCategory	string	SaCategory - AC_SaCategory, see Options: Conditional mandatory with EchoTypeID
DataType	string	Acoustic data type - AC_DataType, see Options
DataUnit	string	Acoustic data unit - AC_DataUnit, see Options
DataValue	decimal	Acoustic data value
CruiseLocalID	string	Unique national cruise identifier

Table 2: Description of dataset Acoustic (Source: https://www.ices.dk/data/Documents/Acoustic/ICES_Acoustic_data_format_description.zip)

Biotic dataset:

	Data Type	Description
Haul	string	Key field used to identify record type
Header/Record	string	Key field used to identify header and record rows
CruiseLocalID	string	Reference to the CruiseLocalID in the Cruise record
HaulGear	string	Biotic sampler - Gear, see Options
HaulNumber	integer	Sequential numbering of hauls during the cruise
HaulStationName	string	Station number. National coding system, not defined by ICES
HaulStartTime	string(16)	Haul start time (GMT) using ISO 8601 format YYYY-MM-DDThh:mm or YYYY-MM-DD hh:mm
HaulDuration	integer	Haul duration in minutes. Start time - the moment when the gear settles at the stated towing speed. Stop is defined as the start of hauling of the gear.
HaulValidity	string	Haul validity code - AC_HaulValidity, see Options
HaulStartLatitude	decimal	Start fishing position: Degree.Decimal Degree of latitude
HaulStartLongitude	decimal	Start fishing position: Degree.Decimal Degree of longitude.
HaulStopLatitude	decimal	Stop fishing position: Degree.Decimal Degree of latitude.
HaulStopLongitude	decimal	Stop fishing position: Degree.Decimal Degree of longitude.
HaulStatisticalRectangle	string	ICES statistical rectangle area reference.
HaulMinTrawlDepth	decimal	Minimum depth (positive value in metres) of the trawl headline. Report only min.depth for the same trawl depth, if different depths applied, report both min. and max. fields
HaulMaxTrawlDepth	decimal	Maximum depth (positive value in metres) of the trawl headline
HaulBottomDepth	decimal	Bottom depth in metres
HaulDistance	integer	Actual distance in metres between haul start and haul end point.
HaulNetopening	decimal	Mean value in metres of vertical net opening measurements
HaulCodendMesh	integer	Codend stretched mesh size in mm
HaulSweepLength	integer	Length of sweep in metres
HaulGearExceptions	string	Gear exceptions - AC_GearExceptions, see Options
HaulDoorType	string	Door type - AC_DoorType, see Options
HaulWarpLength	integer	Length of warp in metres. Defined by fishing depth.
HaulWarpDiameter	integer	Warp diameter in millimetres.
HaulWarpDensity	integer	Warp weight in kg per linear meter of warp.
HaulDoorSurface	decimal	Door surface area in square metres.
HaulDoorWeight	integer	Door weight in kilograms.
HaulDoorSpread	decimal	Mean value in metres of door spread measurements.
HaulWingSpread	decimal	Mean value in metres of wing spread measurements.
HaulBuoyancy	integer	Total buoyancy of the net floats in kilograms.
HaulKiteArea	decimal	Kite area in square metres.
HaulGroundRopeWeight	integer	Ground rope total weight in kilograms.
HaulRigging	string	Rigging is used in the beam trawl surveys.
HaulTickler	integer	Number of ticklers in the Beam trawl surveys.
HaulHydrographicStationID	string	The national hydrographic station reference
HaulTowDirection	integer	Direction of towing in degrees. 360=direction from south to north.
HaulSpeedGround	decimal	Ground speed of towing in knots.
HaulSpeedWater	decimal	Trawl speed on water in knots.
HaulWindDirection	integer	Direction of wind in degrees. Calm=0, 360=direction from north to south.
HaulWindSpeed	integer	Speed of wind in metres/sec.
HaulSwellDirection	integer	Direction of swell in degrees. No movement=0, 360=direction from south to north.
HaulSwellHeight	decimal	Height in metres of the formation of long wavelength ocean surface waves defined as swell
HaulLogDistance	decimal	Distance linking to the acoustic data records
HaulStratum	string	AC_Stratum, see Options

Table 3: Description of dataset Biotic (Source: https://www.ices.dk/data/Documents/Acoustic/ICES_Acoustic_data_format_description.zip)

4.7. Data preparation

The data preparation step is one of the most important and time-consuming steps. Here, data engineering is performed to create new features using the available data. During the data understanding step, the data has already been selected. Following this, the data types of the columns have been set correctly. This is mainly important for the timestamp column since this has to be in a DateTime format to be usable in future steps. As talked about in the previous subsection, LogValidity and HaulValidity determine whether a data entry is valid or not. Therefore, the invalid data entries need to be removed first. This results in a total of 5580758 rows of the acoustic set, losing just 624 data entries. The biotic set has a total of 1876 rows after losing 158 data entries. Furthermore, after retaining only the LogTime, LogLatitude, and LogLongitude columns, duplicates can be removed leaving only 229591 rows for the acoustic set.

The acoustic set will serve as the basis for the features, while the biotic set will serve as the basis for the output variable. However, these two files are not perfectly compatible yet. Therefore, some data preparation is performed to create this match. The acoustic set is used as the basis since this set contains the most data. This set gets a new column called 'Fishing' which can either be 0 (not fishing) or 1 (fishing). To determine which value needs to go in this column, the biotic set is used. A match needs to be found between the timestamp of a haul in the biotic set and the timestamp found in the acoustic set. These matches will be made per CSV file combination, which has gotten a unique file ID in both datasets. Since the latitude and longitude are only known at the acoustic datapoints or the biotic datapoints, the match between the two has to match perfectly. However, the biotic set also contains the HaulDuration column which can be used here. Two situations can occur when assigning the fishing value.

The acoustic timestamp does not have a data entry that falls within the biotic haul window. This biotic haul window is calculated by taking the biotic timestamp and then adding the haul duration to get a start and end timestamp of the haul. If this is the case, then the biotic timestamp including the latitude and longitude of the biotic set will be imported between the two acoustic data entries, as shown in figure 16.

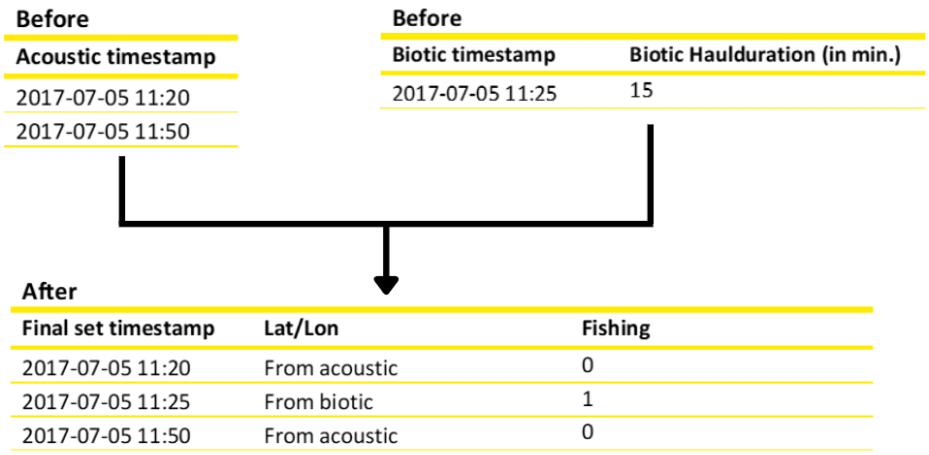


Figure 16: Example 1 of the join between acoustic and biotic

The second situation is when an acoustic timestamp does fall in the biotic haul window. If this is the case the biotic entry gets added as in the first situation and the acoustic timestamp that falls in the biotic haul window gets a 1 value for fishing. This can be seen in figure 17.

Before

Acoustic timestamp
2017-07-05 11:20
2017-07-05 11:30
2017-07-05 11:50

Before

Biotic timestamp	Biotic Haulduration (in min.)
2017-07-05 11:25	15

After

Final set timestamp	Lat/Lon	Fishing
2017-07-05 11:20	From acoustic	0
2017-07-05 11:25	From biotic	1
2017-07-05 11:30	From acoustic	1
2017-07-05 11:50	From acoustic	0

Figure 17: Example 1 of the join between acoustic and biotic

This ensures that the number of fishing locations gets increased which will help with the performance of the model.

With this set, the first few features are engineered and then tested on the models. After optimizing the models for these features and presenting the results to Sustainovate and colleagues within Macaw, feedback has been gathered to add new features. The feature engineering iterations will be discussed in the following subsections.

4.7.1. Iteration 1

For the first iteration of the models, the features have been kept basic and only one additional feature has been engineered. This feature is the speed of the vessel at the current timestamp. This is engineered by calculating the geographical distance between the latitude and longitude at that timestamp and the latitude and longitude of the next timestamp and dividing this by the time between these timestamps. The speed feature might prove crucial in separating non-fishing activities from fishing activities since the shipping vessel has a slower movement speed when fishing than when it is simply going from point A to B.

4.7.2. Iteration 2

During the next iteration of the model, a couple more features were engineered. Since the previous iteration calculated the speed between the current point and the next point, it is true to say that the calculation shows the speed *between* the two points and not per se *at* the point, as can be seen in figure 18.

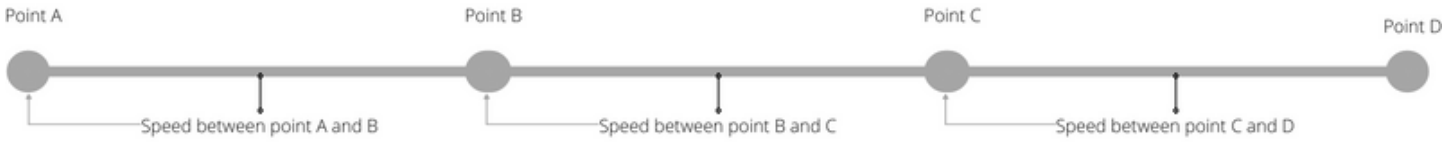


Figure 18: Visual of the speed calculation

Therefore, for iteration two, the calculated speed has been averaged over the lagging speed taking, resulting in a better approximation of the speed at the actual point, as can be seen in figure 19.

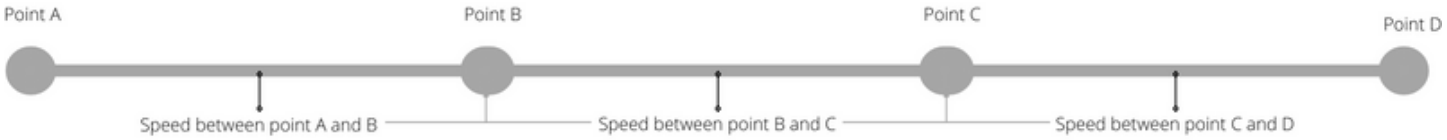


Figure 19: Visual of the speed calculation including lag

This calculation has been done for a single lag, as shown in the above visual, but also with a lag two points down, and a lead of one and two points up. Also, a combination of both a lag and a lead has been included. Finally, the DateTime value has been added as a feature by converting it to Epoch time, meaning the number of seconds since the first of January 1970. This feature is therefore now an integer instead of a DateTime.

4.7.3. Iteration 3

In iteration three the change in bearing has been added. This is one of the key features discussed with Sustainovate alongside the speed feature. The bearing change has been calculated similarly to the speed with lag. First, the bearing has been determined between the current point and the next point. After that, the difference between the bearing to the next point and the bearing from the previous point has been calculated. This is visualized in figure 20.

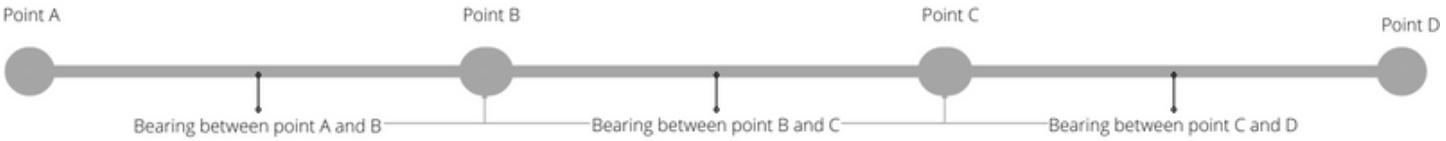


Figure 20: Visual of the bearing change calculation

4.7.4. Iteration 4

For the final iteration, the last feature has been added. This feature is the “distance to port”. For this feature, the “world ports” dataset has been used. From this dataset, only the portname, latitude, and longitude columns have been used to determine the location of the port. To get to the feature, for each point on the vessel's route, the distance to every known port has been calculated. Then, the shortest distance is selected and used as the feature value. The main idea behind this feature is to limit the number of false positives in and around the ports since the values for the speed and bearing change are somewhat similar in these areas as they are when the vessels are fishing.

4.7.5. Segmented analysis

The code to transform the data for the single-point analysis has been used as a basis for creating the segmented dataset. Here, after the features for the single points are calculated, a specific time window has been selected over which the features are aggregated. In the case of the training data, this time window has a duration of 30 minutes since the average fishing period is about 30 minutes. All data entries that fall within a 30-minute time window will get aggregated to an average speed, average bearing over that period, and all position information will be added to a list. This aggregated information will then serve as a single value on which the model will perform the training and testing. If in this 30-minute window an entry has the activity of fishing, the entire 30-minute window will be classified as fishing. Table 4 shows the transformation that takes place using dummy data.

Before

Timestamp	Lat / Lon	Fishing	Speed
2017-07-05 11:20	50 / 20	0	16 km/s
2017-07-05 11:30	51 / 21	0	16 km/s
2017-07-05 11:40	52 / 22	1	7 km/s
2017-07-05 11:45	53 / 23	1	7 km/s
2017-07-05 12:00	54 / 24	0	16 km/s

After

Positions	Window_IDN	Fishing	Average speed
[50/20], [51/21], [52/22], [53/23]	1	1	11,5 km/s
[54/24]	2	0	16 km/s

Table 4: Before and after of segmented transformation

4.8. Modeling

The other big step during the development cycle is model development. The algorithms used for the models are Random Forest and Naïve Bayes. These models are applied to the Single-Point Analysis dataset as well as the Segmented Analysis dataset. Furthermore, each model has been run for each iteration of features.

For the Random Forest model, a grid search is used to tune the hyperparameters with a focus on precision to reach the most optimal model. The parameters and options that are put into the grid search can be seen in table 5.

Parameter	Value 1	Value 2	Value 3	Description
n_estimators	100	250	500	This determines how many trees are in the forest. More trees mean better performance but also slower code.
criterion	Gini	Entropy	Log_Loss	This determines the function that is used to measure how well the split between classes is.

Table 5: Used Hyperparameters for the grid search of the Random Forest Classification

After the first few parameters are known, the max_depth parameter is tested. This determines how far the nodes can expand until all leaves of the tree are pure. When setting this value to 'None', an unlimited amount of nodes can be created, often resulting in overfitting of the model. To test where the model starts to overfit, a range of max_depth values between 1 and 50 is chosen with steps of 4 in between. The model is then trained and tested on the train set and the test set. As soon as the performance of the test set starts to decline it is clear that the model is starting to overfit and the

maximum value for max_depth is reached. This test is done for each iteration using the parameters that came out of the grid search.

For the Support Vector Machines, the only parameter that has been looked at within a grid search is the kernel function. The options used here are the polynomial, radial basis, and sigmoid function. All other parameters have been kept at default.

For the Naïve Bayes, no hyperparameter tuning is possible. Since this model is more focused on speed and less on precision, it became very clear that the imbalance in classes provides an issue for this classifier. Therefore, an under sampler was tested as well.

As a final test, the Auto ML option within azure is used. This Auto ML option will train, test, and compare different models for a selected amount of time to be able to provide the user with the variables of an optimal model. This is used to quickly test models that have not been tested in the custom code. For this test, only the final iteration of features is used, and the primary performance metric is the weighted precision score. All available models have been provided to test and the maximum duration for the test to run is 6 hours.

4.9. Evaluation

After the models had been developed an evaluation is performed by comparing the models with new testing data. The main metric that was used is precision since one of the goals of this research is to provide a model that will have as close to zero false positives as possible. Other performance metrics such as accuracy, F1 score, and recall were looked at as well since they provide some interesting information as well. Also, the feature importance was looked at to see which features contribute the most. The results were then discussed among colleagues and with Sustainovate to come up with improvements for an iteration.

4.10. Deployment and Feedback

This research project is aimed to provide a theoretical foundation and a proof of concept to determine whether implementing such a data pipeline and machine learning model would be beneficial. The deployment step is dependent on the choices made based on this research report and the performance of the proof of concept. Therefore, the deployment and feedback stage will not take place during this research project.

4.11. Concluding thoughts

During this chapter four different data science methodologies were compared after which the Foundational Methodology for Data Science (FMDS) has been applied to the current project. Three different data sources were used. The first data source is the ICES website. They have a databank containing all sorts of surveys of research fishing vessel. Their data consists of a csv-file containing the GPS-data of the vessel and another csv-file containing the catch data of the vessel. This results in 230.215 datapoints of GPS-data and 2.030 datapoints of catch data. The second data source is a validation set containing only AIS-data of a fishing vessel of the coast of Oman, which has 13.281 datapoints. Lastly, a dataset containing 3.582 ports of the world has been used. The data preparation step resulted in two datasets, a single point dataset and a segmented dataset, containing seven features. These datasets where then used on a Support Vector Machine, Naïve Bayes Model, a Random Forest Classifier, and on Azure auto ML before being evaluated using the performance metrics mentioned in Chapter 3. The next chapter will show the results and discuss the performance of these models.

5. Results

In this chapter, the results of all variations of models will be discussed by looking at their performance. The performance metrics that will be looked at are Precision, Recall, F1-score, and accuracy. At the end, a validation dataset of Oman will be used to visualize the predicted fishing locations.

5.1. Single-point analysis

The results of the single-point analysis can be split up into three models. These models can then be split up again in the iterations containing different features. For all models and iterations, a train-test split of 70%-30% respectively has been taken. Furthermore, the split has been stratified over the fishing classes ensuring an identical distribution of the classes in the train and the test set. Lastly, a random state of 899370109 has been used to keep the train test splits similar across different runs.

As a first result, the outcome of the optimal parameters for the random forest classifier based on the grid search can be found in table 6 alongside the result of the optimal max_depth test. The graphs displaying the full max_depth tests can be found in figure 28 in appendix 7.

Iteration	N_estimators	Criterion	Max_depth
Iteration 1	100	Entropy	25
Iteration 2	100	Log_Loss	29
Iteration 3	100	Gini	29
Iteration 4	500	Gini	21

Table 6: Optimal parameters per iteration for RFC single-point

Furthermore, for both the Naïve Bayes and the Support Vector Machines under-sampling was used. The train-test split was first performed, after which the train set has been under-sampled. The test set remained untouched since the environment in which the model will be deployed will also have imbalanced data. After under-sampling the balance of 157367 entries of 'not fishing' and 4768 entries of 'fishing' in the train set, changed to 4768 to 4768.

The kernel function used for the Support Vector Machines can be found in table 7 below. All other parameters were set to the default values as can be found in the documentation (https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html).

Iteration	Full dataset Kernel function	Under sampled dataset Kernel Function
Iteration 1	Radiant Basis Function	Polynomial
Iteration 2	Sigmoid	Polynomial
Iteration 3	Sigmoid	Polynomial
Iteration 4	Sigmoid	Polynomial

Table 7: Optimal kernel functions for SVM single-point

The results of these models can be found in table 10 two pages down.

5.2. Segmented analysis

The results for the segmented analysis will be split up into three models and four iterations as well. Also here, a train-test split of 70%-30% respectively has been taken. The split has been stratified over the fishing classes as well and the random state of 899370109 has been used. As a first result, the outcome of the optimal parameters for the random forest classifier based on the grid search can be found in table 8. Again, the graphs for the max_depth test can be found in figure 28 in appendix 7.

Iteration	N_estimators	Criterion	Max_depth
Iteration 1	100	Gini	5
Iteration 2	250	Gini	9
Iteration 3	100	Log_Loss	5
Iteration 4	100	Gini	9

Table 8: Optimal parameters per iteration for RFC segmented

Again, under-sampling for the train set was performed for the Naïve Bayes and Support Vector Machine. This changed the balance of 914 entries of 'not fishing' and 128 entries of 'fishing' in the train set to 128 to 128.

The kernel functions used for the Support Vector Machines can be found in table 9. Again, all other parameters were set to the default values as can be found in the documentation (https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html).

Iteration	Full dataset Kernel function	Under sampled dataset Kernel Function
Iteration 1	Radiant Basis Function	Radiant Basis Function
Iteration 2	Radiant Basis Function	Radiant Basis Function
Iteration 3	Radiant Basis Function	Polynomial
Iteration 4	Radiant Basis Function	Radiant Basis Function

Table 9: Optimal kernel functions for SVM segmented

The results of these models can be found in table 11 two pages down.

Iteration	Precision	Recall	F1-Score	Balanced Accuracy
Random Forest Classifier				
Iteration 1	78,8%	61,6%	69,2%	80,6%
Iteration 2	86,8%	74,8%	80,3%	87,2%
Iteration 3	87,2%	73,6%	79,9%	86,6%
Iteration 4	86,1%	73,8%	79,4%	86,7%
Naïve Bayes				
Iteration 1	11,8%	5,6%	7,6%	52,2%
Iteration 2	0,0%	0,0%	0,0%	50,0%
Iteration 3	0,0%	0,0%	0,0%	50,0%
Iteration 4	0,0%	0,0%	0,0%	50,0%
Naïve Bayes under-sampled				
Iteration 1	14,8%	16,6%	15,7%	56,9%
Iteration 2	3,2%	72,4%	6,2%	53,5%
Iteration 3	3,2%	72,4%	6,2%	53,5%
Iteration 4	3,2%	72,4%	6,2%	53,5%
Support Vector Machine				
Iteration 1	97,6%	2,0%	3,9%	51,0%
Iteration 2	0,1%	0,1%	0,1%	49,1%
Iteration 3	0,1%	0,1%	0,1%	49,1%
Iteration 4	0,1%	0,1%	0,1%	49,1%
Support Vector Machine under-sampled				
Iteration 1	100,0%	1,4%	2,8%	50,7%
Iteration 2	3,1%	75,0%	6,0%	52,6%
Iteration 3	3,1%	75,0%	6,0%	52,6%
Iteration 4	3,1%	75,0%	6,0%	52,6%

Table 10: Performance metrics for the single-point analysis models

Iteration	Precision	Recall	F1-Score	Balanced Accuracy
Random Forest Classifier				
Iteration 1	84,6%	80,0%	82,2%	89,0%
Iteration 2	89,1%	89,1%	89,1%	93,8%
Iteration 3	90,7%	89,1%	89,9%	93,9%
Iteration 4	87,0%	85,5%	86,2%	91,8%
Naïve Bayes				
Iteration 1	81,8%	32,7%	46,8%	65,9%
Iteration 2	87,5%	76,4%	81,6%	87,4%
Iteration 3	87,0%	72,7%	79,2%	85,6%
Iteration 4	86,7%	70,9%	78,0%	84,7%
Naïve Bayes under-sampled				
Iteration 1	77,8%	50,9%	61,5%	74,4%
Iteration 2	86,4%	69,1%	76,8%	83,8%
Iteration 3	78,0%	58,2%	66,7%	77,9%
Iteration 4	78,6%	60,0%	68,0%	78,9%
Support Vector Machine				
Iteration 1	80,0%	80,0%	80,0%	88,6%
Iteration 2	87,5%	89,1%	88,3%	93,7%
Iteration 3	85,7%	65,5%	74,2%	82,0%
Iteration 4	100,0%	16,4%	28,1%	58,2%
Support Vector Machine under-sampled				
Iteration 1	54,6%	96,4%	69,7%	92,6%
Iteration 2	69,2%	98,2%	81,2%	96,0%
Iteration 3	71,4%	18,2%	29,0%	58,6%
Iteration 4	29,3%	49,1%	36,7%	66,3%

Table 11: Performance metrics for the segmented analysis models

5.3. Feature importance

Calculating the feature importance can be done for the Random Forest Classifier and the Gaussian Naïve Bayes. However, due to the poor performance of the Naïve Bayes model, only the Random Forest will be used to determine feature performance. The feature importance was similar between the single-point dataset and the segmented dataset and can be seen in figure 21.

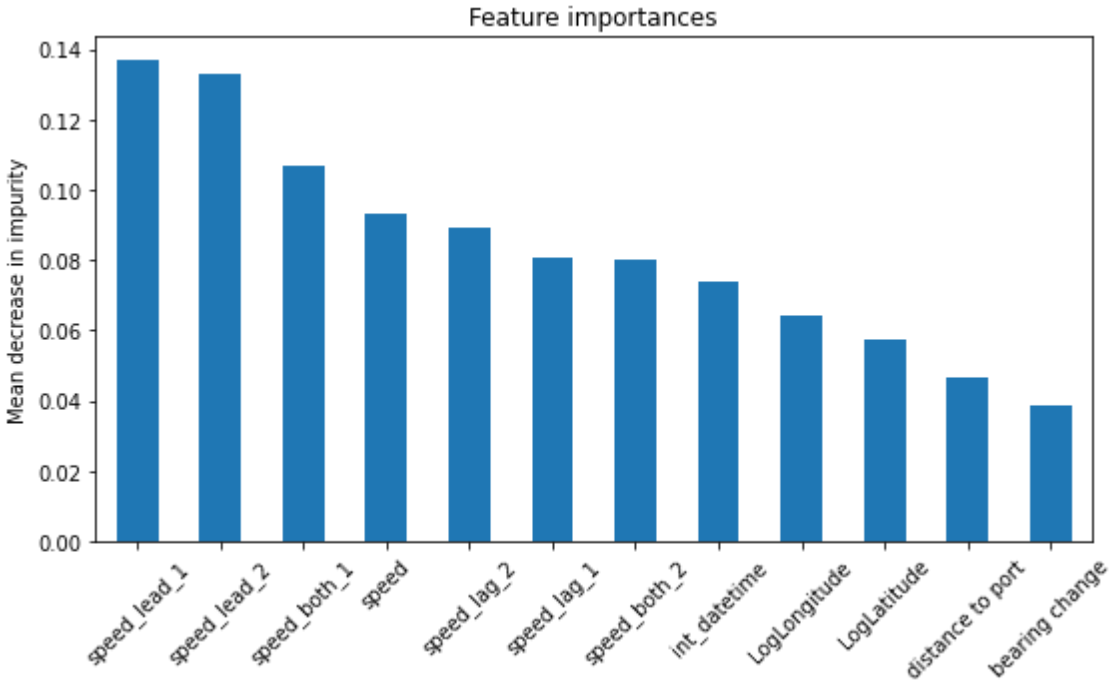


Figure 21: Feature importance based on the Random Forest model

The Auto ML model also calculated the feature importance. These are somewhat different from the ones from the Random Forest Classifier, however, similar conclusions can be drawn. Figure 22 shows the feature importance given by the Auto ML model.

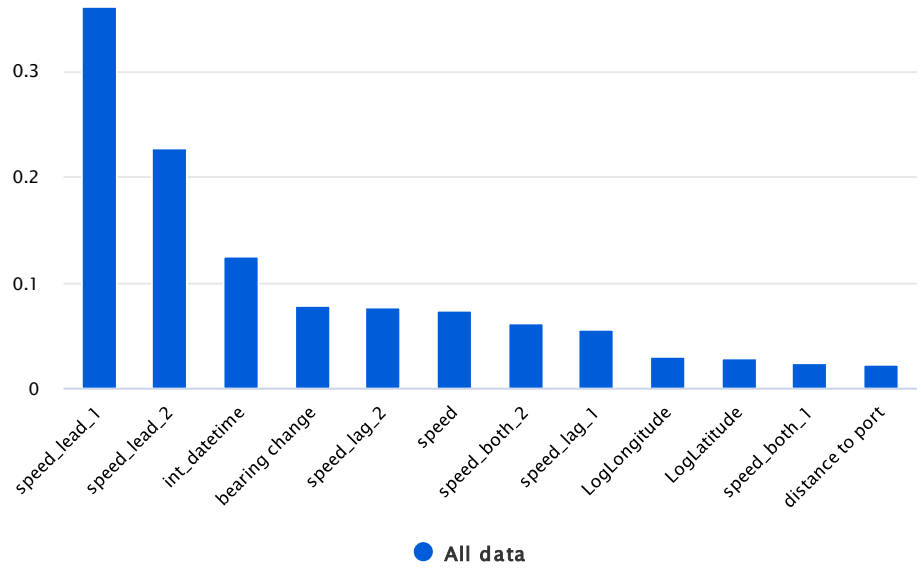


Figure 22: Feature importance based on the Auto ML model

The main differences between the feature importance graphs is that the Random Forest model values all the speed features very highly, indicated by having all of them in the front. The Auto ML model shows something different. While both start with speed_lead_1 and speed_lead_2, the Auto ML model then shows int_datetime and bearing change. Especially the bearing change is interesting since this is the least important feature for the Random Forest model.

5.4. Error analysis

During the error analysis, the goal is to understand why the models make the mistakes they do. The first approach taken to investigate this is to calculate some basic statistics of the model outcome. This has been done for one version of each model that has somewhat acceptable performance, totaling four models. The basic statistics that have been calculated are upper bound, lower bound, median, and average. In table 12, the median value is displayed. The conclusions that will be drawn from the median value could also be drawn from the upper bound, lower bound, and average values since these compare almost identically to each other as the median value does.

Classification type	Median Speed (m/s)	Median Speed_lead_1 (m/s)	Median Bearing Change
Random Forest Classifier (Single-point Iteration 3)			
True Positive	6,9	6,8	2,3
False Positive	6,8	6,6	3,2
True Negative	18,5	18,6	0,5
False Negative	8,0	7,7	6,2
Random Forest Classifier (Segmented Iteration 3)			
True Positive	8,0	7,6	13,6
False Positive	8,0	8,9	9,3
True Negative	18,9	18,6	0,2
False Negative	13,7	12,6	18,4
Naïve Bayes (Segmented Iteration 2)			
True Positive	7,5	7,3	N.A.
False Positive	2,2	9,0	N.A.
True Negative	18,9	18,7	N.A.
False Negative	11,4	11,1	N.A.
Support Vector Machine (Segmented Iteration 2)			
True Positive	8,0	7,7	N.A.
False Positive	17,2	9,8	N.A.
True Negative	18,9	18,7	N.A.
False Negative	13,7	12,6	N.A.

Table 12: Median comparison for error analysis best performers

Table 12 gives the first couple of insights into why certain errors are happening. For the RFC Single-point Iteration 3, the false positives are clearly almost identical to the false positives across all three features, this showing a clear reason as to why the false positives happen on a data level. The false negatives on the other hand are less clear here. The two speed features do appear to be slightly higher than the values for the positive classes, however, it is still much closer to the positive classes than to the negative classes. Furthermore, the bearing change is very high for the false negatives when compared to the true negatives, but also compared to the true positives.

The RFC segmented iteration 3 shows the same result, however, here the speed features are more in the middle between true positive and true negative. The bearing change could be used as a clear indicator as to why this value should be classified as a positive class.

The Naïve Bayes results show similar logic. The speeds of the false positives are closer to the speeds of the true positives than to the true negatives, and the speeds of the false negatives are somewhere in the middle between the true positives and the true negatives.

The Support Vector Machine results are also showing similar results. However, here the ‘speed’ feature is much closer to the true negative value than to the true positive value. The ‘speed_lead_1’ value, on the other hand, shows the opposite, and given the higher feature importance of ‘speed_lead_1’ than that of ‘speed’ the classification will urge to go to positive.

Based on these statistics it is clearer why certain data entries are classified incorrectly. This explanation only works on a data level and still does not tell anything about why the data looks the way it does. To investigate that further, figures 23 and 24 have been developed. Based on these images that visualize the errors no clear conclusion can be drawn why these errors happen. The green dots represent the true positives, the red dots are the false positives, and the blue dots are the false negatives. True negatives have not been displayed since they cover almost the entire route. Both false positives and false negatives almost always happen close to true positives and no clear indicator can be seen to determine when an error of either kind will occur. This single visualization is representative of all tested routes.

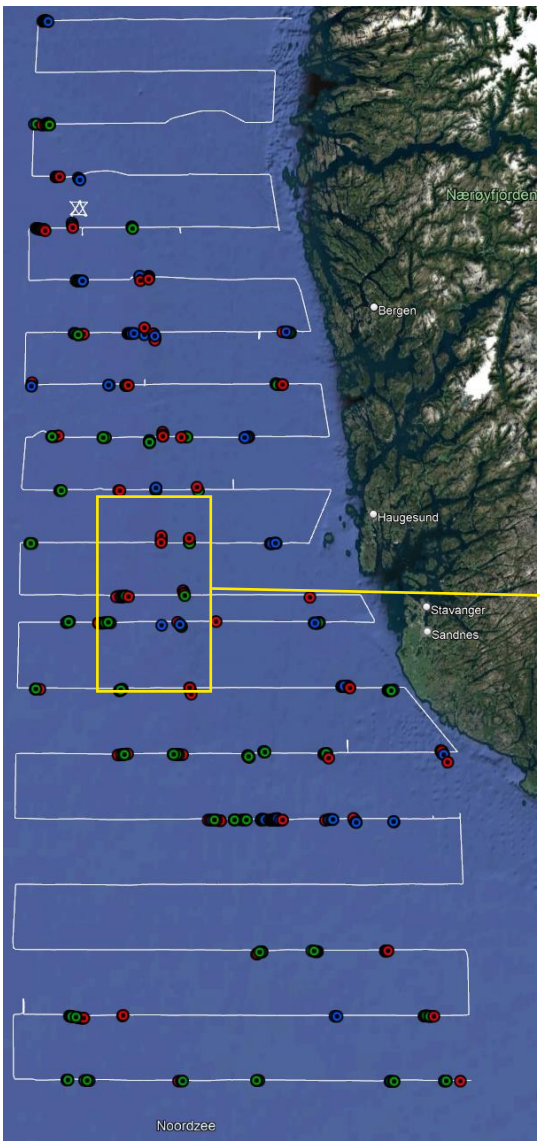
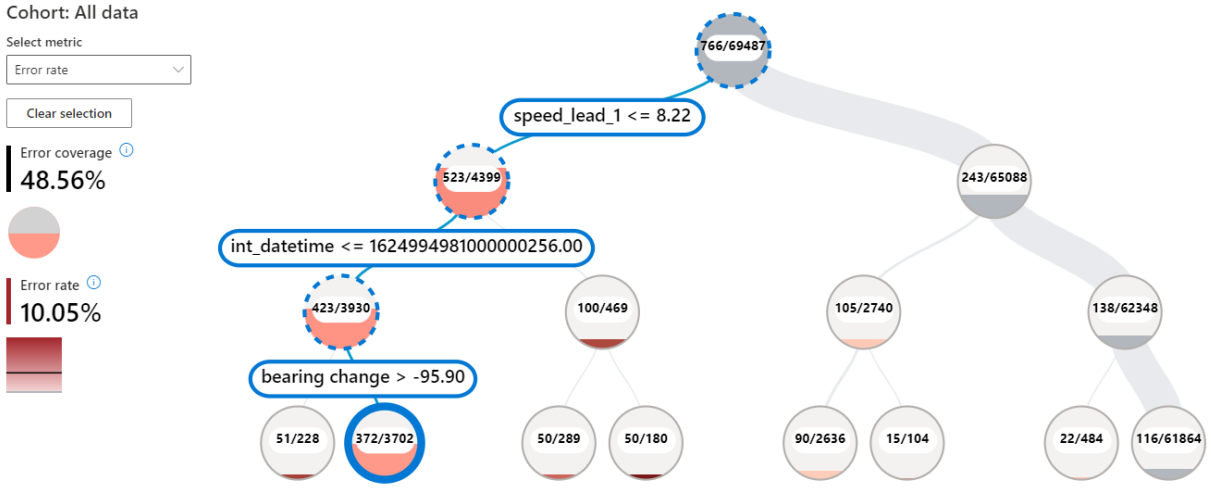


Figure 23: Example of route with true positive (green), false positive (red), false negative (blue)

Figure 24: Example of route with true positive (green), false positive (red), false negative (blue) - zoomed in

Another method used to better understand the errors is by developing a error analysis tree that displays the decision rules that lead to errors. This tree is unique as opposed to a decision tree that shows the decisions rules that leads to a classification, which will be discussed in the chapter 5.5. This error analysis tree highlights the decision rules that lead to an error and thus has no relation to the decision tree in the next paragraph. The decision rules in this tree are created in a hierarchical manner, meaning that the decision rule at the top has the biggest influence on creating errors. This is done with the single-point dataset with iteration 3 in the Random Forest Classifier. Figure 25 shows the decision tree with the line highlighted to get to the highest error coverage. As can be seen, the first decision rule is whether speed_lead_1 is lower or equal to 8.22 m/s. In total this sample contained 766 errors of which 523 errors are present in the lower or equal to 8.22 m/s speed_lead_1 node. This shows that 68.28% of all errors fall within this range. Furthermore, looking further down



this line an error coverage of 48.56% gets reached by having the int_datetime being below 29-06-2021 19:29:41 (if converted from epoch to datetime), and having a bearing change higher than -95.90 degrees. Using these decision rules we encounter the highest number of errors in a single node with an error rate of 10.05%.

Figure 25: Error analysis tree view, highest error coverage node. Figure created using the Error Analysis Dashboard from the "Responsible AI Toolbox" called Raiwidgets in python. Each node indicates the number of errors in the total numbers of datapoints, showcased by "errors/total datapoints". Each connection between the nodes is a new decision rule that leads to a new "error/total datapoints". These decision rules indicate the best borders to segment the data in errors and correct predictions. The error coverage on the left displays the percentage of all error in the dataset concentrated in the selected node. The error rate displays the percentage of failures of all the datapoints in the selected node. (https://pypi.org/project/raiwidgets/)

Looking at figure 26 shows the highest error rate within this tree. An error rate of 27.78% with an error coverage of 6.53%. The decision rules to get here are speed_lead_1 lower or equal to 8.22 m/s, datetime of beyond 29-06-2021 19:29:41, and a LogLatitude higher than 61.24. The neighboring node has an error rate of 17.30% with an error coverage of 6.53% where only the LogLatitude decision rule changes to being lower or equal to 61.24. Both of these nodes have a very high error rate with a combined error coverage of 13.05% and error rate of 21.32%. This shows that data entries which have a spead_lead_1 lower than or equal to 8.22 m/s and have been executed past 29-06-2021 19:29:41 are very prone to being an error.

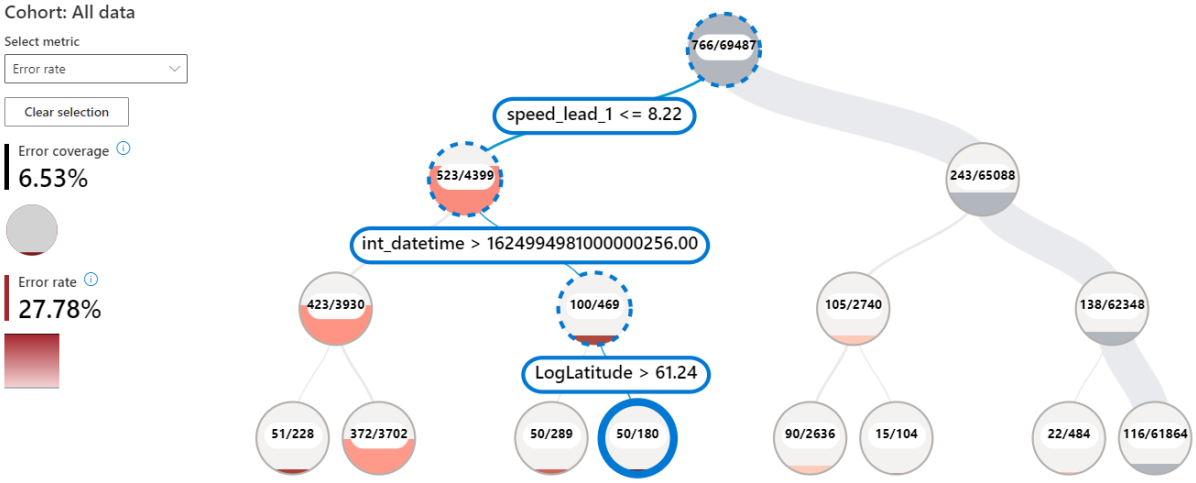


Figure 26: Error analysis tree view, highest error rate node. Figure created using the Error Analysis Dashboard from the "Responsible AI Toolbox" called Raiwidgets in python.

The right side of the tree contains 31.72% of all errors. However, it also contains 93.67% of all data entries. This indicates that the vast majority of errors (68.28%) occur in a very small minority of data (6.33%), represented by the left side. The main takeaway from this is that all entries with a speed_lead_1 of less or equal to 8.22 m/s are difficult to predict. When looking exclusively at the false positives 198 out of 210 false positive, thus 94.29% , fall within this range. Since the goal is to reduce the false positives as much as possible, it becomes clear that this range is something to focus on when improving the model. When visualizing these specific values using Google Earth, a similar figure as figure 23 and 24 from two pages earlier gets created. This figure does not provide any additional insights.

5.5. Explainability

One of the greatest advantages of the classic machine learning models used during this research is the fact that they are white box models, meaning it is possible to take a closer look into the decision making process of the model. This has been done for the Random Forest Classifier using the Segmented dataset with iteration 3. Figure 27 shows a decision tree of the model. This decision tree gives us insights in the decision rules made throughout the tree. All the way on top the first decision rule is whether the speed is smaller or equal to 12.12 m/s. If this is true the sample goes to the left to the next decision rule. In this next decision rule at 'value = [75, 104]' and 'class = Fishing' it can be seen that the majority (104) of the samples are of class 'Fishing'. This indicates that if the speed is smaller or equal to 12.12 m/s then the sample is most likely a fishing sample. From here the next decision rule asks the sample whether it has a bearing change smaller or equal to -190.91 degrees. If this is the case the sample gets shifted to the node to the left and down. This node has no new decision rule since all samples in this node have the same class, fishing. This shows that if a sample has a speed of less or equal to 12.12 m/s and a bearing change of less or equal to -190.91 degrees than the sample will be a fishing sample.

Using this decision tree some hypothesis can be stated. First of, it seems that a lower speed mostly leads to 'Fishing' classifications. However, as can be seen from the fourth decision rule from the top on the left side, if the speed falls below 8.26 m/s the balance between fishing an not fishing classifications is 50/50, indicated by the only white box. This shows that while a lower speed usually leads to a 'Fishing' classification, a speed that is too low could very well also lead to a 'Not Fishing' classification. A similar conclusion can be made about bearing change, indicating that a bearing change closer to zero more often leads to a 'Not Fishing' classification, while a bearing change further away from zero more often leads to a 'Fishing' classification. These rules are not surprising given that fishing vessels need to have a lower speed to fish and will have to turn more to fish the same area multiple times to catch most fish present in that area. Also, when the speed becomes too low the fishing vessel might be stationary and is thus not fishing, indicating that the decisions the tree makes are recognizable to what happens in reality.

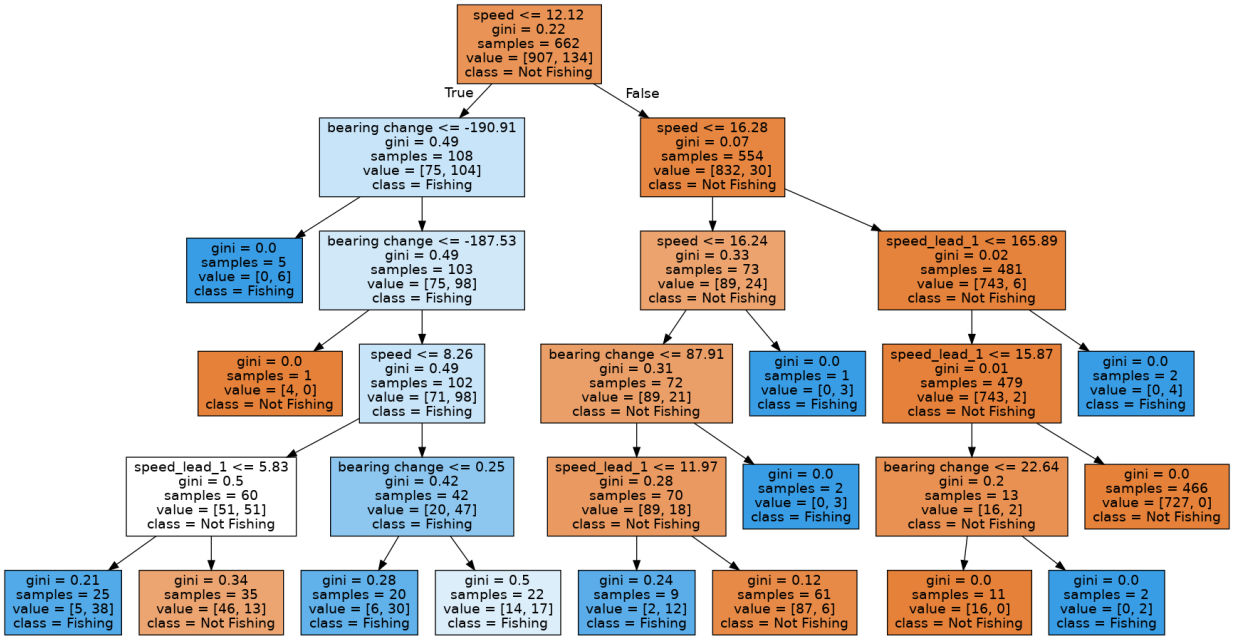


Figure 27: A decision tree of the Random Forest Classifier using SDA iteration 3. Figure created using the scikit-learn package in python. For each box, the top line indicates the decision rule applied, the second line indicates the criterion and criterion value used, the third line shows the amount of samples used, the fourth line shows the distribution of classes (this is different to the number of samples due to bootstrapping), the fifth line shows the main class in the box. The color of the box indicates the class, orange for 'Not fishing', blue for 'Fishing'. The darker the color the more dominant the class.

5.6. Auto ML

Lastly, the results of the Auto ML run can be shown. As said before, the Auto ML function has been run only over the final iteration, thus containing all available features. After running for 1,5 hours, no more improvements over the last 20 models were detected thus the run was ended. The model with the best performance was the Voting Ensemble model. This model consisted of 8 ensemble algorithms which are detailed in table 13. The detailed parameters used for these ensemble algorithms can be found in tables 23 through 31 in appendix 8.

Ensemble algorithm	Data transformation	Training algorithm	Weight
1	Sparse Normalizer	XGBoost Classifier	0,23
2	Sparse Normalizer	XGBoost Classifier	0,15
3	Standard Scaler Wrapper	XGBoost Classifier	0,15
4	Sparse Normalizer	XGBoost Classifier	0,15
5	Sparse Normalizer	XGBoost Classifier	0,08
6	Sparse Normalizer	XGBoost Classifier	0,08
7	Truncated SVD Wrapper	Random Forest Classifier	0,08
8	Max Abs Scaler	LightGBM	0,08

Table 13: Overview of ensemble algorithms in best performing Auto ML model

For this model, the entire dataset is used, thus no under-sampling is done resulting in imbalanced data as input. The results of this model can be seen in table 14.

Metric	Score
Precision	95,6%
Recall	91,1%
F1-Score	93,3%
Balanced accuracy	91,1%

Table 14: Performance metrics of the Auto ML model

5.7. Oman verification

To validate the performance of the model outside of its own dataset, the Oman dataset is used to predict vessel activity and to plot this in Google Earth. This visual representation can then be used to discuss the performance with Sustainovate. The training model used to predict the Oman dataset is from the segmented analysis, iteration 2 of the Support Vector Machine model without under-sampling. This is referenced as model 1. Since this model is trained on the segmented dataset, a second model is used. This model is iteration 2 of the Random Forest Classifier of the single-point analysis. This is referenced as model 2.

Figure 29 in appendix 9 shows the prediction of the SVM model laid out over the route the vessel has taken. Here, the red lines are the parts where the model classified the vessel to be fishing and the white lines are the route the vessel has taken. In figure 30 in appendix 9, a closeup of the port area can be seen. Here, the red lines are the parts where the model classified the vessel to be fishing and the white lines are the route the vessel has taken. The predictions of the second model, the Random Forest, can be seen in figure 31 in appendix 10. Figure 32 in appendix 10 shows the closeup of the port for the second model.

Figures 33 and 34 in appendix 11 show the result GFW showed for this vessel during this period. Looking at the legend in the bottom right corner it would seem that a lot of fishing is being predicted since the lines have the same color as the “fishing” option. However, if it was truly a fishing prediction it should have been a dot with that color and not the line. This is an unfortunate visualization issue with the map. However, what it shows is that the GFW model does not predict any fishing locations for this specific vessel even though it is known that the vessel was fishing during this period.

5.8. Performance of the Global Fishing Watch

A crucial comparison to make is between the performance of the newly developed models and the performance of the current model of GFW. As mentioned in the problem description, the product GFW has developed shows some false positives which indicate a below 100% precision, however, the full performance has not been discussed yet. One of the researchers at GFW was kind enough to give access to their research paper which contains the same performance metrics as used for the other models. The results from this paper for the trawlers can be found in table 15. The data GFW used for their research was AIS data from 2012-2016 provided by ORBCOMM, as well as using data from Souze et al. (2016, as cited in, Kroodsma et al., 2018). In total, they had over 247,000 hours of AIS tracks, including over 569,000 data points with a fishing or not-fishing label. No information is known about the balance of the classes. In total they used twelve features (Kroodsma et al., 2018):

- The difference in timestamp between current and previous point.
- Distance in meters between current and previous point.
- Current reported speed.
- Implied speed by the first two features.
- Change in course over ground between current and previous point.
- The local time.
- Current month.
- Change in course over ground between current, previous, and next point.
- Distance to the nearest shore.
- Distance to the nearest anchorage.
- Time to the nearest anchorage visit.
- Number of vessels within 1 kilometer radius.

Metric	Score
Precision	98%
Recall	94%
F1-Score	96%
Accuracy	96%

Table 15: Performance metrics of GFW's model for trawlers (Kroodsma et al., 2018, p30)

5.9. Concluding thoughts

After running the models on the test set and validating them using the Oman set, some first conclusions can be made. Given the performance metrics alone, four models stand out as decent performers. These are the Random Forest Classifier using the single-point dataset iteration 3, Random Forest Classifier using the segmented dataset iteration 3, Naïve Bayes using the segmented dataset iteration 2, and Support Vector Machine using the segmented dataset iteration 2. Also, the Auto ML model has great performance. These models, however, still do not outperform the model of GFW. Looking at the error analysis, a bit more insight can be gathered about why certain things are being classified wrong. 68% of all errors occur when the speed_lead_1 is below 8.22 m/s. When looking at the false positives, almost 95% occur within this range. This indicates that predicting a vessels activity when their speed_lead_1 drops below 8.22 m/s is very difficult.

To learn more about the decision making process of the models a single decision tree taken from a Random Forest is investigated. The decisions made by the tree are similar to the way a fishing vessel operates as discussed with Sustainovate. The tree shows that when speed is relatively low and bearing change is relatively far away from zero, the vessel is most likely fishing, while a speed that is too low might again lead to a ‘Not Fishing’ classification. This is in line with reality where fishing

vessels need to lower their speed and change direction often to empty out a specific area of fish, and an even lower speed could indicate a (almost) stationary vessel, which is thus not fishing.

Using these results, it becomes possible to start to answer the research questions set in the first chapter. Therefore, the next chapter will draw conclusion on these results and answer the research questions, while also reflecting on the entire process of this research.

6. Conclusion & Reflection

In this chapter, the results and literature combined with the expert knowledge and set goals of Sustainovate will be used to answer the research questions set at the beginning of this thesis and to provide insights into a possible solution to the problem.

The goal of this research is to provide Sustainovate with a white box machine learning model that can classify fishing vessel activities using AIS-data. This classification can then be used to serve as a proxy in a different model that predicts the location and size of fish biomass. This will be used to help fisheries in improving their fishing strategies and to aid governing agency in their decision-making. Besides these benefits, the fishing vessel activity classification model can be used to detect illegal fishing. The Global Fishing Watch currently has a free to use product that provides fishing vessel activity classifications, however, this product is build upon a Convolutional Neural Network, which is a black box model. This has as a drawback that the decision-making process of the model is very difficult to interpret, and thus users just have to trust that the model is correct. Besides that, their model shows numerous false positives which should be minimized. This research has therefore investigated possibilities to create a model that is explainable while having similar to better performance. To get to this goal a couple of research questions have been developed. To be able to formulate a better solution to the goal, these research questions will first be discussed.

6.1. Research questions

The research questions were as follows and will be discussed in greater detail in the following paragraphs:

1. *How can individual GPS/AIS-datapoints be pre-processed to be used for predictive modeling of activity classification?*
    - *1.1. What techniques can be used to group individual GPS/AIS-datapoints into classified events?*
    - *1.2. How can GPS/AIS-based events of different lengths be used together in machine learning models?*
2. *To what extent can supervised classification algorithms help in detecting fishing vessel activities, specifically fishing and non-fishing, based on GPS/AIS data?*
    - *2.1. What features can be identified to help predict fishing vessel activity?*
    - *2.2. What is the performance difference between various supervised classification algorithms in predicting fishing vessel activities?*
3. *How can the final machine learning model results contribute to the OceanBox fish mapping forecasting service?*

6.1.1. Research question 1

Research question 1 has a strong focus on the data preprocessing part of model development. The answers to questions 1.1 and 1.2 are not as straightforward as the questions might indicate. Where the questions would fit an answer that simply explains different techniques, the reality shows something different. During the start of this research, the hypothesis was that for the model to be useful, it would have to be able to detect the start and finish of when a vessel is doing one single activity. However, after diving deeper into the literature and understanding the available data, it became clear that this was not the case. Instead of being able to know when the activity starts and finishes, it is important to know what a vessel is doing at a specific time. This means that the individual data points do not need to be grouped into classified events. This simultaneously means that the problem of having events of different lengths will no longer be an issue. However, one of the methods used during this research is a segmented data analysis, which does group individual data points together. This grouping, however, is not based on the underlying event but is based on a

specific set time period. Therefore, to answer research question 1: Individual GPS/AIS-datapoints can be pre-processed as individual instances or by grouping them in time-restrained segments and aggregating the features.

6.1.2. Research question 2

For research question 2, it is more useful to first answer the two sub-questions since clear answers can be given to these. So, during this research, multiple iterations of features have been developed. These features are, DateTime since epoch, latitude, longitude, speed, speed with a lead up to two, speed with a lag up to two, speed with a combination of a lead and a lag up to two, bearing change, and distance to port. After running the models, the feature importance of these features was calculated, resulting in a clear indication of the features that are most important to predicting vessel activity. While all features add some value, some features truly stood out as important predictors. These features are speed and speed_lead_1 when looking at the RFC feature importance. Speed_lead_1 is the most important feature for the Auto ML feature importance as well, while speed_lead_2 is the runner-up. This can be seen in the feature importance figures (figure 24 and figure 25) but also by looking at the changes in performance across iterations. In most models, a large jump in performance can be seen between iterations 1 and 2. The changes in performance for iterations 3 and 4 are small. Since iterations 3 and 4 contain the features 'bearing change' and 'distance to port', it can be deduced that these features are not as important as the previous features.

While the features are an important factor in model performance, so is the model itself. During this research three different models have been developed, using two types of data sources. The Random Forest Classifier is very consistent between the single-point dataset and the segmented dataset, showing similar performance across iterations. The Support Vector Machine and Naïve Bayes models, on the other hand, show a large difference in performance between the single-point dataset and the segmented dataset. Both models perform much better using the segmented dataset. By under-sampling the dataset, a decrease in performance can be noted. This is most likely because only the trainset is under-sampled and not the test-set since in production the data would also be imbalanced. Based on these values, the conclusion can be drawn that under-sampling the data does not improve performance.

Lastly, the Auto ML run shows amazing performance. Here, a voting ensemble consisting of 8 classifiers, as can be found in table 13, outperforms all other tested models on all metrics using the same, not under-sampled, data. Most interestingly, the high performance of an ensemble model indicates that using multiple algorithms simultaneously might be the optimal approach. More tests need to be run, including a test for multicollinearity, however.

Given these conclusions, it can be stated that using supervised classification algorithms can prove to be highly useful in predicting fishing vessel activities using AIS/GPS data.

6.1.3. Research question 3

Figures 29 through 32 (Appendix 9 & 10) give a first indication of the usability of these models for Sustainovate and the Oceanbox environment. These figures help in showcasing the performance of the models when confronted with a dataset that closely resembles what would be used within the OceanBox environment. The Oman dataset has a higher level of detail meaning more data points per time period are gathered. Also, the duration of the fishing activities is longer, and the data has been gathered during the full fishing season, instead of just a very small portion of this. A few things can be concluded from these figures, which have been discussed with Sustainovate. First off, the area around the port has been an issue in the model of GFW, showing fishing activities even though they

are not allowed here. In their current version of the model, this has been fixed to a certain degree. As can be seen in figures 33 and 34 (Appendix 11) a new activity called 'port visit' has been added, which for this particular vessel has been predicted well. Issues here, however, is that their model does not predict any fishing activities for this vessel. This means that it is unclear whether all false positive fishing activities in the ports are classified as port visits. While a fishing activity could still happen in that area, almost all predicted activities in that area are false positives. Both models developed during this research have not managed to improve on these issues. Figures 30 and 32 (Appendix 9 & 10) clearly show multiple predicted fishing activities in the port area. What both models also show, which can be seen in figures 29 and 31 (Appendix 9 & 10), is that three distinct fishing areas can be seen. These areas are indicated by the high concentration of red lines or dots. After diving deeper with the Founder & General Manager of Sustainovate and looking into these areas and the areas connecting them, the conclusion could be drawn that it makes sense that, given the shapes of the white lines, those locations would be the fishing hotspots. One important side note here is that the red dots mostly indicate the turning points of the fishing vessel, while the actual fishing is being done between the red dots. The red lines, indicating data from the segmented dataset, showcases this better since those lines represent the fishing track.

These findings result in both new insights as well as new questions. The most basic conclusion that can be drawn from these findings is that the contribution these machine learning models provide to the OceanBox environment is that in the current state these models do not provide additional benefits compared to the model of GFW. However, the models developed during this research are usable to an extent, indicating that with further development they might provide the additional benefit for the OceanBox environment that this research was looking for. By looking at the error analysis a first clear indication can be made as to where the biggest improvements can be made. The error analysis clearly shows that almost 95% of all false positives occur when the speed_lead_1 is or drops below 8.22 m/s. Besides that, 68% of all errors occur within this range, while this range only contains 6.33% of all data. By finding a solution to improve the accuracy within this specific range, the overall performance of the model can be improved on drastically. Currently, too little is known as to why this range specifically is providing difficulties.

6.2. Research goal

Based on the results of the research questions, a better understanding has been developed as to how to reach the research goal. As mentioned in the introduction of this chapter, the goal is to develop an explainable white box model with similar to better performance when compare with the black box model of GFW. Research questions 3 has made a start to providing insights into this goal. The models developed have not matched the performance of the model of GFW, however, they are explainable. Investigating a decision tree shows a clear logic within the decision-making process of the model. When a vessel is going relatively fast and making very little changes in direction the model predicts the vessel to not be fishing. When a vessel is going relatively slow and is making larger changes in direction, the model predicts the vessel to be fishing. Lastly, when the vessel is going even slower, the model predicts the vessel to not be fishing. As discussed with Sustainovate, this logic is sound when compared to reality. The fact that this logic is sound and can be demonstrated and explained to potential clients of Sustainovate, might give Sustainovate a competitive advantage when compare to the model of GFW. The white box nature of the model can provide the clients with a higher level of trust concerning the outcome of the model. This is both useful for fisheries wanting to improve their strategies as well as governing agencies that intend to use the model to combat illegal fishing. By being able to back up their conclusion that someone is illegally fishing, with the arguments given by the white box model, opens up a better discussion than would be possible when these arguments are not available.

6.3. Reflection

Working on this thesis has had its ups and downs, as any such project will have. During this research multiple parties have been reached out to in order to work together on this project. One of these parties is the Global Fishing Watch. Their fishing vessel activity classification model has been a center piece during this entire research. Said model has been published publicly on their Github. At the beginning of this research the goal was to use a dataset on their open source model and then compare it to my own developed models. This would ensure a perfect one-to-one comparison of the models. However, I failed to get the model to run on my own device. After multiple conversations and mail correspondence with the employees of GFW we all failed to detect the underlying problem as to why the model did not run on my device. Some final suggestions were made which were not in reach for me to try unfortunately. This resulted in not being able to run the one-to-one comparison. The workers at GFW were kind enough to provide me with their own research paper, which had been developed alongside the model development. This paper contained their performance metrics, enabling me to still be able to make somewhat of a comparison, even though it is far from a perfect one.

Furthermore, during the earlier stages of this research conversations were held with parties that were in possession of valuable AIS and catch data. These datasets contained a very high level of detail and would be able to provide me with data that strongly resembles the data sources used by GFW and the level of detail needed for the final product as suggested by Sustainovate's OceanBox. After an initial proposal these parties seemed very keen on working with us, however, as time progressed this enthusiasm did unfortunately not lead to any further collaboration. This initial enthusiasm, however, does show their interest in the subject. This thesis might help mitigate any doubts these parties might have, especially since no strict deadline would be present when starting a follow-up project, possibly resulting in collaboration later on.

Even though these events did not lead to any collaboration, the time invested of these parties to hear us out and respond to our requests has been very much appreciated. Furthermore, the alternative solutions, such as the research paper of GFW and the datasets of ICES, have enabled me to develop this thesis. While the comparisons that can be made using these alternative solutions are not as strong, the research approach developed in this thesis is sound and can be used as a guide to redo this research or develop a similar one. None the less, the development of this thesis has enabled me to grow in the data science field and taught me new things along the way.

7. Discussion

This chapter will look further into the theoretical contributions that can be gathered from this research as well as the managerial implications the conclusion might have. Also, the limitations of this research will be discussed before making recommendations for future research.

7.1. Limitations

To start, the limitations will be discussed. Often, this sub-section is discussed after the theoretical contributions and managerial implications, however, for this research, it makes sense to put this first. The reason for this is that the limitations this research has, are shown to have a very large effect on the results of this research. The biggest and most important limitation of this research is the ICES dataset. While this dataset provided a large quantity of data on many fishing vessels, each CSV file contained the same crucial flaws. The first is the approach taken to capture the data. The data points for the acoustic dataset have been registered for each nautical mile. This results in a variable granularity of data. When the vessel is going fast, the level of detail is high since the nautical mile is reached in a short period, but when the vessel is going slow, the level of detail is low for the reverse reason. This is a crucial limitation in the data for the aimed goals of this research. Fishing vessels are driving slower when fishing than when they are going from A to B, resulting in less information available during the fishing periods.

Another limitation caused by the data is the fact that the fishing periods were very short. The average fishing duration calculated from the HaulDuration column in the biotic dataset is 35 minutes. This does not represent an actual fishing duration of multiple hours very well. While this is not a big issue, combined with the low level of detail due to the GPS data being available every nautical mile, results in a lot more missing important information.

The final limitation of the dataset has to do with the ports. After adding the ‘distance to port’ feature, somewhat of a performance improvement was expected. Especially when plotting the predicted fishing locations of the Oman dataset. The expected result was to have limited the number of false positives in the port areas, however, this was not the case. This triggered an investigation in the ICES dataset, and after plotting several routes of this data it became clear that this feature would indeed add no value. The limitation found, was that the ICES data capture was started when the vessel was already at sea and stopped before returning to port. This results in no available information around the port area. Therefore, making it impossible for the models to learn that the closer a vessel gets to the port, the less likely it is that that vessel is fishing, which the ‘distance to port’ feature was meant to enable.

Another limitation was the inability to get the publicly available code for the Neural Network model from GFW to run. This could have proven useful in comparing their model to the newly developed models. By testing their performance on the ICES data as well.

The final limitation is the absence of catch data for the Oman dataset. While it is very useful to be able to have a dataset to use for validation, the validation method is far from perfect. Since there was no catch data available a perfect one-on-one comparison between the predicted fishing locations and the actual fishing locations was not possible. The alternative method, looking and guessing whether the outcomes ‘make sense’ to an expert, is the best method available given the circumstances, but certainly not a statistically sound one. Also, the validation is done for a single vessel during a single year. This specific vessel and year seem to perform very badly for the GFW model. It is, however, difficult to draw strong conclusions from a comparison of a single entity.

7.2. Theoretical contributions

Given the results and the conclusions, and the limitations, the main theoretical contribution this research provides is that it showcases the effect of having a dataset for training a model that is not perfectly in line with the prospected data to be used to work with the model. After this research, a lot of questions concerning the usability of the developed models remain, mainly due to the mismatch in training and validation data.

Another contribution this research provides is showcasing the effect segmenting the data in larger periods than just a single point can have. The results show that for the Naïve Bayes and the Support Vector Machine the performance between the single-point dataset and the segmented dataset are very different, whereas the segmented dataset performs much better. The Random Forest Classifier, however, is not much affected by the difference in dataset, indicating that this is not a generalizable rule for all machine learning algorithms.

7.3. Managerial implications

The managerial implications of the research can be split up into two parts. The first part regards the outcomes of this research as final, and the second part regards the outcomes as a first step. So, when looking at these results, the conclusion for Sustainovate is to use data coming from GFW as opposed to developing their custom model. The model of GFW outperforms the custom models developed during this research and is thus more reliable to use.

However, when using the outcomes as a first step to developing a custom in-house model, then this research could be used as a guide to improving the developed models. By learning from the mistakes showcased in the limitations and making use of the suggestions made in future research, the models can be improved, and a couple of benefits can be gained from this.

First off, when having this improved model, the output can be gathered on-demand. Sustainovate can request the output at any time and can alter the information that is outputted when requirements change. Since the goal for the OceanBox is to use the results from the models developed in this research as a proxy for their Biomass estimation model, a constant feed of data can be achieved by setting up some data pipelines. Furthermore, more control over the output also means more uses of the output. The results that are provided now are aimed to be very precise with regards to detecting the exact location where a vessel is fishing. This precise information can be used for fishery inspections by showing the exact location a vessel has been fishing. However, this information can also be used on an aggregate level. By using these fishing locations, a polygon or heatmap can be drawn showcasing the prospected presence of fish. If a vessel is fishing right at the border of a protected area, circling it, then both outputs can be provided. The exact fishing locations can be displayed, to determine whether the vessel was fishing inside or outside the protected area. But also, a heatmap can be drawn, most likely showcasing a high presence of fish within the circle, indicating a lot of fish within the protected area.

This shows that the model can have uses for different applications. These two applications will also require a different granularity of data, since the fish presence is an approximation, thus getting by at a lower granularity, while the exact fishing location of a vessel requires a high granularity since this information needs to be precise.

Furthermore, with some more work the classical machine learning models might be able to achieve similar performance as the deep learning CNN model. Two major benefits to the classical machine learning models are that they are computationally much cheaper and are easier to explain. When using the outcomes of the models to support fisheries or convince governing bodies it is crucial to be

able to explain why the outcomes are the way they are. Using a classical machine learning model makes this much easier to explain to a non-technical individual than when using a deep learning model.

7.4. Future research

The limitations subsection has provided some clear opportunities for improvement, thus calling out for future research. The biggest opportunity for future research is to redo this research with a new dataset containing a higher level of detail and a better match between the training data and the data that will be used after the model would be deployed. The hypothesized result of this would be a model that outperforms the model of GFW thus enabling the benefits mentioned in the previous subsection. This dataset would need to register information at a consistent time interval, for example, every minute. Furthermore, the data should be gathered anytime the vessel is moving, thus including the parts where it is coming in and going out of the port. Also, the data should be gathered during a longer period of fishing. The catch data should match the actual catch times as well, which, for as far as known, the ICES dataset did. A suggestion would be to receive the data from a large AIS-data supplier such as MarineTraffic (https://www.marinetraffic.com/nl/ais-api-services) or Spire (https://spire.com/maritime/) and to get a fishery on-board to provided their catch data.

Furthermore, this research has focused on one specific vessel type, the trawler. It would be interesting to research the effects of adding different vessel types and sizes. Also, the training data from ICES contained data mostly gathered from the North Sea, therefore it would be interesting to research the effect a different geographical location has.

One interesting attribute of the ICES dataset that has not been used during this research, is the extensive amount of biological data available in the biotic data. This information contains a lot of information about the specific fish that has been caught and could be used in future research to make predictions about the type of fish that is been fished for.

Also, using a Fourier Transformation on the data could enable the development of a “fingerprint” of an activity. This type of analysis might improve the performance of classifying fishing vessel activities. It might also enable or improve the detection of differences between types of vessels or detect the difference between fish types that are being fished for.

References

Ahmed, H., & Nandi, A. K. (2019), "Frequency Domain Analysis," in Condition Monitoring with Vibration Signals: Compressive Sampling and Learning Algorithms for Rotating Machines , IEEE, pp.63-77, doi: 10.1002/9781119544678.ch4.

Ataspinar, A. (2018, 12 april). Machine Learning with Signal Processing Techniques. ML Fundamentals. Accessed on 23 March 2022, van https://ataspinar.com/2018/04/04/machine-learning-with-signal-processing-techniques/

Bergès, B. J. P., Sakinan, S., Berg, F., Lusseau, S. M., Schaber, M., & O'Connnell, S. (2021). HERAS survey indices: automation, TAF and testing (No. 21.007). Stichting Wageningen Research, Centre for Fisheries Research (CVO).

Browning, E., Bolton, M., Owen, E., Shoji, A., Guilford, T., & Freeman, R. (2018). Predicting animal behaviour using deep learning: GPS data alone accurately predict diving in seabirds. Methods in Ecology and Evolution, 9(3), 681-692.

Cannon, J. (2020, 3 februari). Illegal industrial fishing hampers small-scale African fisheries. Mongabay Environmental News. https://news.mongabay.com/2020/02/illegal-industrial-fishing-hampers-small-scale-african-fisheries/

Cardoso-Fernandes, J., Teodoro, A. C., Lima, A., & Roda-Robles, E. (2020). Semi-automatization of support vector machines to map lithium (Li) bearing pegmatites. Remote Sensing, 12(14), 2319.

Data Science Process Alliance. (2021, 24 august). CRISP-DM. https://www.datascience-pm.com/crisp-dm-2/

Dietterich, T. G. (2000). Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Springer, Berlin, Heidelberg.

Dike, H. U., Zhou, Y., Deveerasetty, K. K., & Wu, Q. (2018, October). Unsupervised learning based on artificial neural network: A review. In 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS) (pp. 322-327). IEEE.

Eddy, S. R. (2004). What is a hidden Markov model?. Nature biotechnology, 22(10), 1315-1316.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM, 39(11), 27-34.

Foroughi, F., & Luksch, P. (2018). Data science methodology for cybersecurity projects. arXiv preprint arXiv:1803.04219.

Gitis, V. G., Derendyaev, A. B., & Petrov, K. N. (2021, September). Earthquake Prediction Based on Combined Seismic and GPS Monitoring Data. In International Conference on Computational Science and Its Applications (pp. 601-612). Springer, Cham.

Global Fishing Watch. (n.d. a). About us. https://globalfishingwatch.org/about-us/

Global Fishing Watch. (n.d. b). Datasets and code. https://globalfishingwatch.org/data-download/datasets/public-fishing-effort

Global Fishing Watch. (n.d. c) https://globalfishingwatch.org/marine-protected-areas/

Global Fishing Watch. (n.d. d). Transshipment. https://globalfishingwatch.org/transshipment/

Global Fishing Watch. (n.d. e). What Vessels are requires to use AIS? What are global regulations and requirements for vessels to carry AIS? https://globalfishingwatch.org/faqs/what-vessels-are-required-to-use-ais-what-are-global-regulations-and-requirements-for-vessels-to-carry-ais/

Global Fishing Watch. (n.d. f). Commercial Fishing. https://globalfishingwatch.org/commercial-fishing/

Gutierrez-Torre, A., Berral, J. L., Buchaca, D., Guevara, M., Soret, A., & Carrera, D. (2020). Improving maritime traffic emission estimations on missing data with CRBMs. Engineering Applications of Artificial Intelligence, 94, 103793.

ICES. (2015) Manual for International Pelagic Surveys (IPS). Series of ICES Survey Protocols SISP 9 – IPS. 92 pp.

Kontopoulos, I., Chatzikokolakis, K., Tserpes, K., & Zissis, D. (2020, July). Classification of vessel activity in streaming data. In Proceedings of the 14th ACM International Conference on Distributed and Event-based Systems (pp. 153-164).

Kroodsma, D. A., Mayorga, J., Hochberg, T., Miller, N. A., Boerder, K., Ferretti, F., ... & Worm, B. (2018). Tracking the global footprint of fisheries. Science, 359(6378), 904-908.

Liu, A. Y. C. (2004). The effect of oversampling and undersampling on classifying imbalanced text datasets (Doctoral dissertation, University of Texas at Austin).

Mandal, J. K., & Bhattacharya, D. (2020). Emerging Technology in Modelling and Graphics. Springer Singapore.

Marine Institute, Wageningen Marine Research, Institue of Marine Research, PINRO, Faroese Marine Research Institute, Marine Sctoland Marine Laboratory, Johann Heinrich von Thünen-Insitut, Danish Institute for Fisheries Research, Galway/Mayo Institute of technology, & Irish Parks and Wildlife Service. (2018). INTERNATIONAL BLUE WHITING SPAWNING STOCK SURVEY (IBWSS) SPRING 2018. https://oar.marine.ie/handle/10793/1349

Marine Traffic. (z.d.). What is the significance of the AIS Navigational Status Values? https://help.marinetraffic.com/hc/en-us/articles/203990998-What-is-the-significance-of-the-AIS-Navigational-Status-Values-

Marktab, Alexbuckgit, & V-kent (Github handles). (2021a, 12 November). What is the Team Data Science Process. Microsoft. https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview

Marktab, Alexbuckgit, & V-kent (Github handles). (2021b, 12 November). The business understanding stage of the Team Data Science Process lifecycle. Microsoft. https://docs.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle-business-understanding

Marktab, Alexbuckgit, & V-kent (Github handles). (2021c, 15 December). Data acquisition and understanding stage of the Team Data Science Process lifecycle. Microsoft. https://docs.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle-data

Marktab, EdPrice-MSFT, Alexbuckgit, & V-kent (Github handles). (2021d, 29 December). Modeling stage of the Team Data Science Process lifecycle. Microsoft. https://docs.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle-modeling

Marktab, Alexbuckgit, Igayhardt, & V-kent (Github handles). (2021e, 15 December). Deployment stage of the Team Data Science Process lifecycle. Microsoft. https://docs.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle-deployment

Marktab, Alexbuckgit, & V-kent (Github handles). (2021f, 15 December). Customer acceptance stage of the Team Data Science Process lifecycle. Microsoft. https://docs.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle-acceptance

Munoz-Organero, M., Ruiz-Blaquez, R., & Sánchez-Fernández, L. (2018). Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving. Computers, Environment and Urban Systems, 68, 1-8.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society, 18(6), 275-285.

Nanda, M. A., Seminar, K. B., Nandika, D., & Maddu, A. (2018). A comparison study of kernel functions in the support vector machine and its application for termite detection. Information, 9(1), 5.

Noble, W. S. (2006). What is a support vector machine?. Nature biotechnology, 24(12), 1565-1567.

Nøttestad, L., Anthonypillai, V., Dos Santos Schmidt, T. C., Høines, Å., Salthaug, A., Ólafsdóttir, A. H., Kennedy, J., Jacobsen, J. A., Smith, L., Eliasen, S. K., Jansen, T., & Wieland, K. (2021, August). Cruise report from the international ecosystem summer survey in the nordic seas (IESSNS). https://www.hi.no/resources/WD09-IESSNS_survey_report_2021.pdf

OceanBox. (n.d.). Digital bycatch – A new valuable Resource in Fisheries. https://oceanbox.eu/#section-about

Okey, T. A., Griffiths, S., Pascoe, S., Kenyon, R., Miller, M., Dell, Q., Pillans, R., Buckworth, R. C., Gribble, N., Engstrom, N., Bishop, J., Milton, D., Salini, J., & Stevens, J. (2007). Effects of Illegal Foreign Fishing on the Ecosystem in the Gulf of Carpentaria: Management Options and Downstream Effects on Other Fisheries. CSIRO Marine and Atmospheric Research.

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

Piatetsky, G. (2014, October). CRISP-DM, still the top methodology for analytics, data mining, or data science projects, https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

Qin, T. (2020). Machine Learning Basics. In Dual Learning (pp. 11-23). Springer, Singapore.

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. ieee assp magazine, 3(1), 4-16.

Ren, J., Lee, S. D., Chen, X., Kao, B., Cheng, R., & Cheung, D. (2009). Naive bayes classification of uncertain data. In 2009 Ninth IEEE international conference on data mining (pp. 944-949). IEEE.

Rollins, J.B. (2015). Foundational Methodology for Data Science. IBM Analytics. https://www.ibm.com/downloads/cas/WKK9DX51

Saha, S., & Ekbal, A. (2013). Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. Data & Knowledge Engineering, 85, 15-39.

Saritas, M. M., & Yasar, A. (2019). Performance analysis of ANN and Naive Bayes classification algorithm for data classification. International Journal of Intelligent Systems and Applications in Engineering, 7(2), 88-91.

Scheffer, M., Carpenter, S., & de Young, B. (2005). Cascading effects of overfishing marine systems. Trends in ecology & evolution, 20(11), 579-581.

Seymore, K., McCallum, A., & Rosenfeld, R. (1999, July). Learning hidden Markov model structure for information extraction. In AAAI-99 workshop on machine learning for information extraction (pp. 37-42).

Sicotte, X. B. (2018, 28 June). Kernels and Feature maps: Theory and intuition. https://xavierbourretsicotte.github.io/Kernel_feature_map.html

Singh, K. K., Elhoseny, M., Singh, A., & Elngar, A. A. (Eds.). (2021). Machine Learning and the Internet of Medical Things in Healthcare. Academic Press.

Soares, E. F. D. S., Revoredo, K., Baião, F., de MS Quintella, C. A., & Campos, C. A. V. (2019). A combined solution for real-time travel mode detection and trip purpose prediction. IEEE Transactions on Intelligent Transportation Systems, 20(12), 4655-4664.

Stojov, V., Koteli, N., Lameski, P., & Zdravevski, E. (2018). Application of machine learning and time-series analysis for air pollution prediction. Proceedings of the CIIT.

Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130.

Sustainovate. (n.d. a). The OceanBox – For improved efficiency and responsible fishing. https://sustainovate.com/portfolio/oceanbox/

Sustainovate. (n.d. b). Welcome to our agency. https://sustainovate.com/about/

Suthaharan, S. (2016). Support vector machine. In Machine learning models and algorithms for big data classification (pp. 207-235). Springer, Boston, MA.

Visa, S., Ramsay, B., Ralescu, A. L., & Van Der Knaap, E. (2011). Confusion matrix-based feature selection. MAICS, 710(1), 120-127.

Wang, S. C. (2003). Artificial neural network. In Interdisciplinary computing in java programming (pp. 81-100). Springer, Boston, MA.

Widdicombe, S., Dashfield, S. L., McNeill, C. L., Needham, H. R., Beesley, A., McEvoy, A., Øxnevad, S., Clarke, K. R., & Berge, J. A. (2009). Effects of CO2 induced seawater acidification on infaunal diversity and sediment nutrient fluxes. Marine ecology progress series, 379, 59-75.

Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (Vol. 1, pp. 29-39).

World Food Programme. (z.d.). Who we are. https://www.wfp.org/who-we-are

Xiao, G., Cheng, Q., & Zhang, C. (2019). Detecting travel modes from smartphone-based travel surveys with continuous hidden Markov models. International Journal of Distributed Sensor Networks, 15(4), 1550147719844156.

Zhang, L., Dalyot, S., Eggert, D., & Sester, M. (2011). Multi-stage approach to travel-mode segmentation and classification of gps traces. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences:[Geospatial Data Infrastructure: From Data Acquisition And Updating To Smarter Services] 38-4 (2011), Nr. W25, 38(W25), 87-93.

Appendix

Appendix 1: Example of dataset acoustic

	Example 1	Example 2	Example 3
Data	Data	Data	Data
Header	Record	Record	Record
LogDistance	1021.0	1021.0	1022.0
LogTime	29-6-2017 16:59	29-6-2017 16:59	29-6-2017 17:05
LogLatitude	54,49603	54,49603	54,47940
LogLongitude	7,806822	7,806822	7,808015
LogOrigin	start	start	start
LogLatitude2	NaN	NaN	NaN
LogLongitude2	NaN	NaN	NaN
LogOrigin2	NaN	NaN	NaN
LogValidity	V	V	V
LogBottomDepth	NaN	NaN	NaN
SampleChannelDepthUpper	11.0	12.0	10.0
SampleChannelDepthLower	12.0	13.0	24.0
SamplePingAxisInterval	1.0	1.0	1.0
SamplePingAxisIntervalType	distance	distance	distance
SamplePingAxisIntervalUnit	nmi	nmi	nmi
SampleSvThreshold	-70	-70	-70
InstrumentID	SBT1WBT38	SBT1WBT38	SBT1WBT38
CalibrationID	SB736T138	SB736T138	SB736T138
DataAcquisitionID	SB736DA	SB736DA	SB736DA
DataProcessingID	SB736DP	SB736DP	SB736DP
SamplePingAxisIntervalOrigin	start	start	start
DataSaCategory	CLU	CLU	CLU
EchoTypeID	NaN	NaN	NaN
DataType	C	C	C
DataUnit	m2nmi-2	m2nmi-2	m2nmi-2
DataValue	0.03	2.377.245	0.0
CruiseLocalID	06SL736	06SL736	06SL736

Table 16: Example of dataset acoustic

Appendix 2: Example of dataset biotic

	Example 1	Example 2	Example 3
Haul	Haul	Haul	Haul
Header	Record	Record	Record
CruiseLocalID	06SL736	06SL736	06SL736
HaulGear	PEL	PEL	PEL
HaulNumber	1	10	11
HaulStationName	317	350	353
HaulStartTime	30-6-2017 05:33	5-7-2017 11:25	6-7-2017 07:09
HaulDuration	15	30	10
HaulValidity	V	V	V
HaulStartLatitude	54,10233	52,93800	53,35017
HaulStartLongitude	7,80417	2,53950	2,97817
HaulStopLatitude	54,08833	52,94483	53,34833
HaulStopLongitude	7,80317	2,58417	2,99300
HaulStatisticalRectangle	37F7	34F2	35F2
HaulMinTrawlDepth	34.0	25.0	26.0
HaulMaxTrawlDepth	34.0	29.0	26.0
HaulBottomDepth	43.0	39.0	32.0
HaulDistance	1558.0	3088.0	1005.0
HaulNetopening	6.0	7.0	6.0
HaulCodendMesh	20.0	20.0	20.0
HaulSweepLength	NaN	NaN	NaN
HaulGearExceptions	NaN	NaN	NaN
HaulDoorType	NaN	NaN	NaN
HaulWarpLength	225.0	145.0	145.0
HaulWarpDiameter	NaN	NaN	NaN
HaulWarpDensity	NaN	NaN	NaN
HaulDoorSurface	NaN	NaN	NaN
HaulDoorWeight	NaN	NaN	NaN
HaulDoorSpread	NaN	NaN	NaN
HaulWingSpread	NaN	NaN	NaN
HaulBuoyancy	NaN	NaN	NaN
HaulKiteArea	NaN	NaN	NaN
HaulGroundRopeWeight	NaN	NaN	NaN
HaulRigging	NaN	NaN	NaN
HaulTickler	NaN	NaN	NaN
HaulHydrographicStationID	NaN	NaN	NaN
HaulTowDirection	172.0	95.0	87.0
HaulSpeedGround	3.5	3.4	3.3
HaulSpeedWater	3.9	3.8	3.2
HaulWindDirection	296.0	359.0	123.0
HaulWindSpeed	8.0	2.0	5.0
HaulSwellDirection	NaN	NaN	NaN

HaulSwellHeight	NaN	NaN	NaN
HaulLogDistance	NaN	NaN	NaN
HaulStratum	NaN	NaN	NaN

Table 17: Example of dataset biotic

Appendix 3: Key statistics of the data

Acoustic dataset:

	Empty percentage	Unique values
Data	0%	1
Header	0%	1
LogDistance	0%	129118
LogTime	0%	222401
LogLatitude	0%	146370
LogLongitude	0%	202053
LogOrigin	0%	2
LogLatitude2	100%	0
LogLongitude2	100%	0
LogOrigin2	100%	0
LogValidity	0%	1
LogBottomDepth	37%	133936
SampleChannelDepthUpper	0%	254
SampleChannelDepthLower	0%	23130
SamplePingAxisInterval	0%	520
SamplePingAxisIntervalType	0%	1
SamplePingAxisIntervalUnit	0%	1
SampleSvThreshold	0%	14
InstrumentID	0%	5
CalibrationID	0%	9
DataAcquisitionID	0%	9
DataProcessingID	0%	8
SamplePingAxisIntervalOrigin	3%	2
DataSaCategory	0%	15
EchoTypeID	100%	0
DataType	0%	1
DataUnit	0%	1
DataValue	0%	4237090
CruiseLocalID	0%	76

Table 18: Statistics of dataset Acoustic

Biotic dataset:

	Empty percentage	Unique values
Haul	0%	1
Header	0%	1
CruiseLocalID	0%	70
HaulGear	0%	4
HaulNumber	0%	926
HaulStationName	0%	1122
HaulStartTime	0%	2026
HaulDuration	0%	112
HaulValidity	0%	2
HaulStartLatitude	0%	1893
HaulStartLongitude	0%	1966
HaulStopLatitude	25%	1459
HaulStopLongitude	25%	1518
HaulStatisticalRectangle	48%	386
HaulMinTrawlDepth	0%	459
HaulMaxTrawlDepth	43%	442
HaulBottomDepth	28%	652
HaulDistance	55%	757
HaulNetopening	0%	218
HaulCodendMesh	76%	5
HaulSweepLength	85%	6
HaulGearExceptions	100%	0
HaulDoorType	84%	2
HaulWarpLength	62%	196
HaulWarpDiameter	92%	3
HaulWarpDensity	100%	0
HaulDoorSurface	85%	6
HaulDoorWeight	85%	6
HaulDoorSpread	77%	320
HaulWingSpread	100%	0
HaulBuoyancy	100%	0
HaulKiteArea	100%	0
HaulGroundRopeWeight	88%	4
HaulRigging	100%	0
HaulTickler	100%	0
HaulHydrographicStationID	100%	0
HaulTowDirection	62%	285
HaulSpeedGround	81%	89
HaulSpeedWater	86%	60
HaulWindDirection	79%	177
HaulWindSpeed	81%	21

HaulSwellDirection	98%	9
HaulSwellHeight	98%	10
HaulLogDistance	87%	255
HaulStratum	97%	1

Table 19: Statistics of dataset Biotic

Appendix 4: Example of dataset acoustic after clearing empty columns

	Example 1	**Example 2**	**Example 3**
Data	Data	Data	Data
Header	Record	Record	Record
LogDistance	1021.0	1021.0	1022.0
LogTime	29-6-2017 16:59	29-6-2017 16:59	29-6-2017 17:05
LogLatitude	54,49603	54,49603	54,47940
LogLongitude	7,80682	7,80682	7,80802
LogOrigin	start	start	start
LogValidity	V	V	V
SampleChannelDepthUpper	11.0	12.0	10.0
SampleChannelDepthLower	12.0	13.0	24.0
SamplePingAxisInterval	1.0	1.0	1.0
SamplePingAxisIntervalType	distance	distance	distance
SamplePingAxisIntervalUnit	nmi	nmi	nmi
SampleSvThreshold	-70	-70	-70
InstrumentID	SBT1WBT38	SBT1WBT38	SBT1WBT38
CalibrationID	SB736T138	SB736T138	SB736T138
DataAcquisitionID	SB736DA	SB736DA	SB736DA
DataProcessingID	SB736DP	SB736DP	SB736DP
SamplePingAxisIntervalOrigin	start	start	start
DataSaCategory	CLU	CLU	CLU
DataType	C	C	C
DataUnit	m2nmi-2	m2nmi-2	m2nmi-2
DataValue	0.03	2.377.245	0.0
CruiseLocalID	06SL736	06SL736	06SL736

Table 20: Example of dataset acoustic after clearing empty columns

Appendix 5: Example of dataset biotic after clearing empty columns

	Example 1	Example 2	Example 3
Haul	Haul	Haul	Haul
Header	Record	Record	Record
CruiseLocalID	06SL736	06SL736	06SL736
HaulGear	PEL	PEL	PEL
HaulNumber	1	10	11
HaulStationName	317	350	353
HaulStartTime	30-6-2017 05:33	5-7-2017 11:25	6-7-2017 07:09
HaulDuration	15	30	10
HaulValidity	V	V	V
HaulStartLatitude	54,10233	52,93800	53,35017
HaulStartLongitude	7,80417	2,53950	2,97817
HaulMinTrawlDepth	34.0	25.0	26.0
HaulNetopening	6.0	7.0	6.0

Table 21: Example of dataset biotic after clearing empty columns

Appendix 6: Example of dataset world ports

	Example 1	Example 2
FID	wld_trs_ports_wfp.14314	wld_trs_ports_wfp.14315
portname	Watsi-Genge	Charlotte (Skidegate)
code		CASKI
prttype	River	Sea
prtsize	Very Small	Unknown
status	Unknown	Open
maxdepth		
maxlength		
annualcapacitymt		
humuse	Unknown	Unknown
locprecision	unknown	accurate
latitude	-0,9456	53,24742
longitude	20,62966	-132,00969
iso3	COD	CAN
iso3_op	COD	CAN
country	Democratic Republic of the Congo	Canada
lastcheckdate		
remarks		
url_lca		
source		
createdate	2021-02-24 11:52:47	2021-02-24 11:52:47
updatedate	2021-02-24 11:52:47	2021-02-24 11:52:47
geonameid	204280	6148858
gdb_geomattr_data		
shape	POINT (20.629661913000007 -0.9456024360000015)	POINT (-132.009692535 53.247424029)

Table 22: Example of dataset world ports

Appendix 7: Max_depth graphs for Random Forest parameter tuning

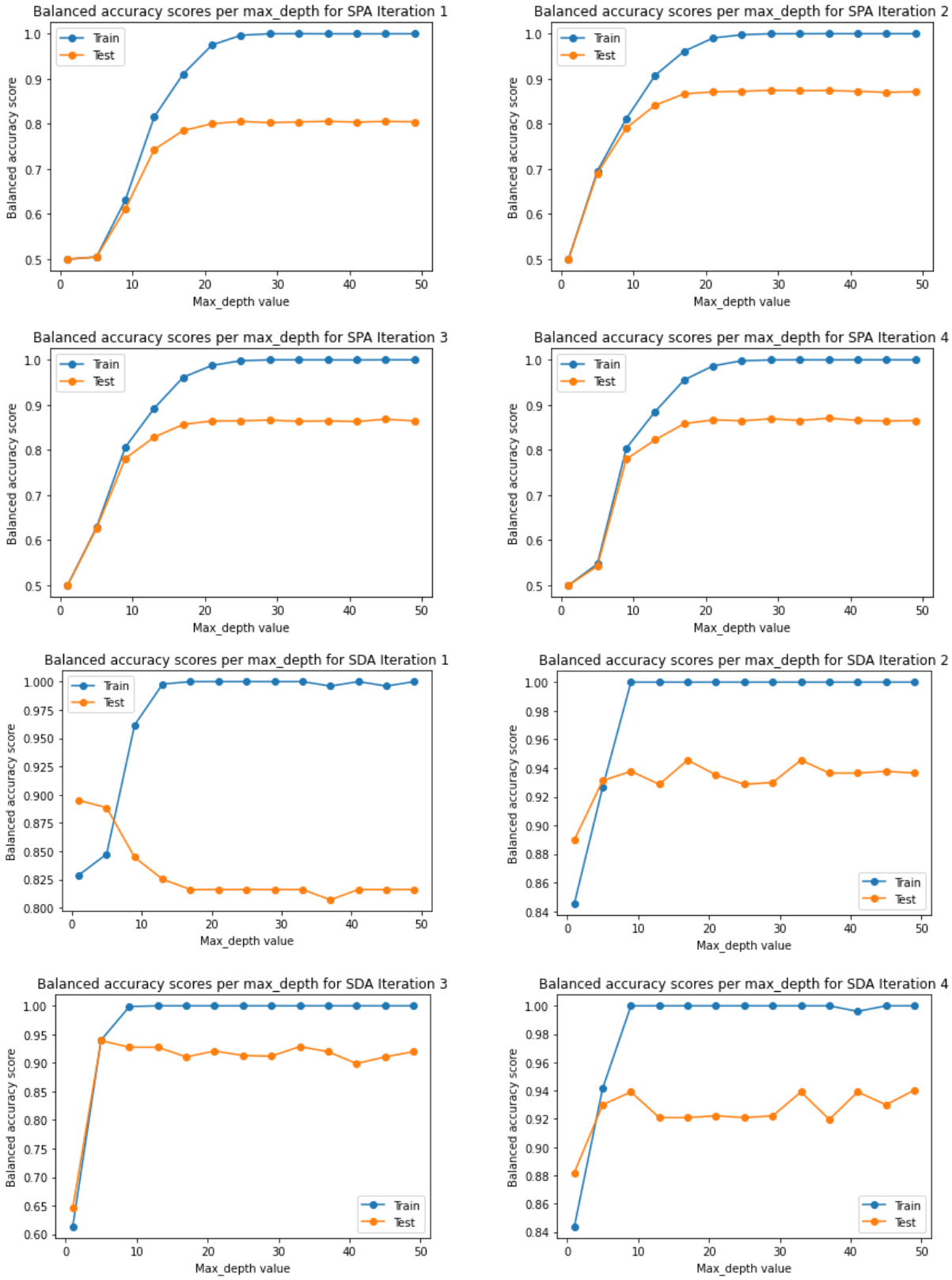


Figure 28: Max_depth graphs for RFC. SPA is Single Point Analysis, SDA is Segmented Data Analysis

Appendix 8: Auto ML parameters

In this appendix you will find all parameters used for the Auto ML ensemble algorithms. Table 20 below is a copy of table 10 to help easily understand what parameters goes with what ensemble algorithm. The 'Ensemble algorithm' column used in the following tables references to the number in table 20.

Ensemble algorithm	Data transformation	Training algorithm	Weight
1	Sparse Normalizer	XGBoost Classifier	0,23
2	Sparse Normalizer	XGBoost Classifier	0,15
3	Standard Scaler Wrapper	XGBoost Classifier	0,15
4	Sparse Normalizer	XGBoost Classifier	0,15
5	Sparse Normalizer	XGBoost Classifier	0,08
6	Sparse Normalizer	XGBoost Classifier	0,08
7	Truncated SVD Wrapper	Random Forest Classifier	0,08
8	Max Abs Scaler	LightGBM	0,08

Table 23: Copy of table 10: Overview of ensemble algorithms in best performing Auto ML model

Sparse normalizer parameters:

Ensemble algorithm	Norm
1	L1
2	L2
4	Max
5	L2
6	L2

Table 24: Sparse normalizer parameters

Standard Scaler Wrapper parameters:

Ensemble algorithm	With_mean	With_std
3	False	False

Table 25: Standard Scaler Wrapper parameters

Truncated SVD Wrapper parameters:

Ensemble algorithm	N_components
7	0,7026

Table 26: Truncated SVD Wrapper parameters:

Max Abs Scaler parameters:

Ensemble algorithm	All parameters
8	No parameters

Table 27: Max Abs Scaler parameters:

XGBoost Classifier parameters part 1:

Ensemble algorithm	Booster	Colsample_ bylevel	Colsample_ bytree	eta	gamma	Max_ depth	Max_ leaves
1	Gbtree	0,5	1,0	0,5	0,0	7	0
2	Gbtree	1,0	1,0	0,4	0,0	5	3
3	Gbtree	0,6	1,0	0,3	0,0	6	0
4	Gbtree	1,0	0,6	0,4	0,0	6	63
5	Gbtree	1,0	0,6	0,3	0,1	6	3
6	Gbtree	1,0	0,9	0,3	0,0	9	0

Table 28: XGBoost Classifier parameters part 1

XGBoost Classifier parameters part 2:

Ensemble algorithm	N_ estimators	Objective	Reg_alpha	Reg_ lambda	Subsample	Tree_ method
1	100	Reg:logistic	0	1,25	1,0	Auto
2	800	Reg:logistic	0	0,00	0,7	Auto
3	100	Reg:logistic	0	1,25	1,0	Auto
4	800	Reg:logistic	0	1,00	1,0	Auto
5	800	Reg:logistic	0	4,00	1,0	Auto
6	200	Reg:logistic	0	1,35	0,8	Auto

Table 29: XGBoost Classifier parameters part 2

Random Forest Classifier parameters:

Ensemble algorithm	Bootstrap	Class_weight	Criterion	Max_ features	Min_ samples_ leaf	Min_ samples_ split	N_ estimators	Oob_ score
7	False	Balanced	Gini	Log2	0,01	0,01	200	False

Table 30: Random Forest Classifier parameters

LightGBM parameters:

Ensemble algorithm	Min_data_in_leaf
8	20

Table 31: LightGBM parameters

Appendix 9: Visualization of the validation of model 1

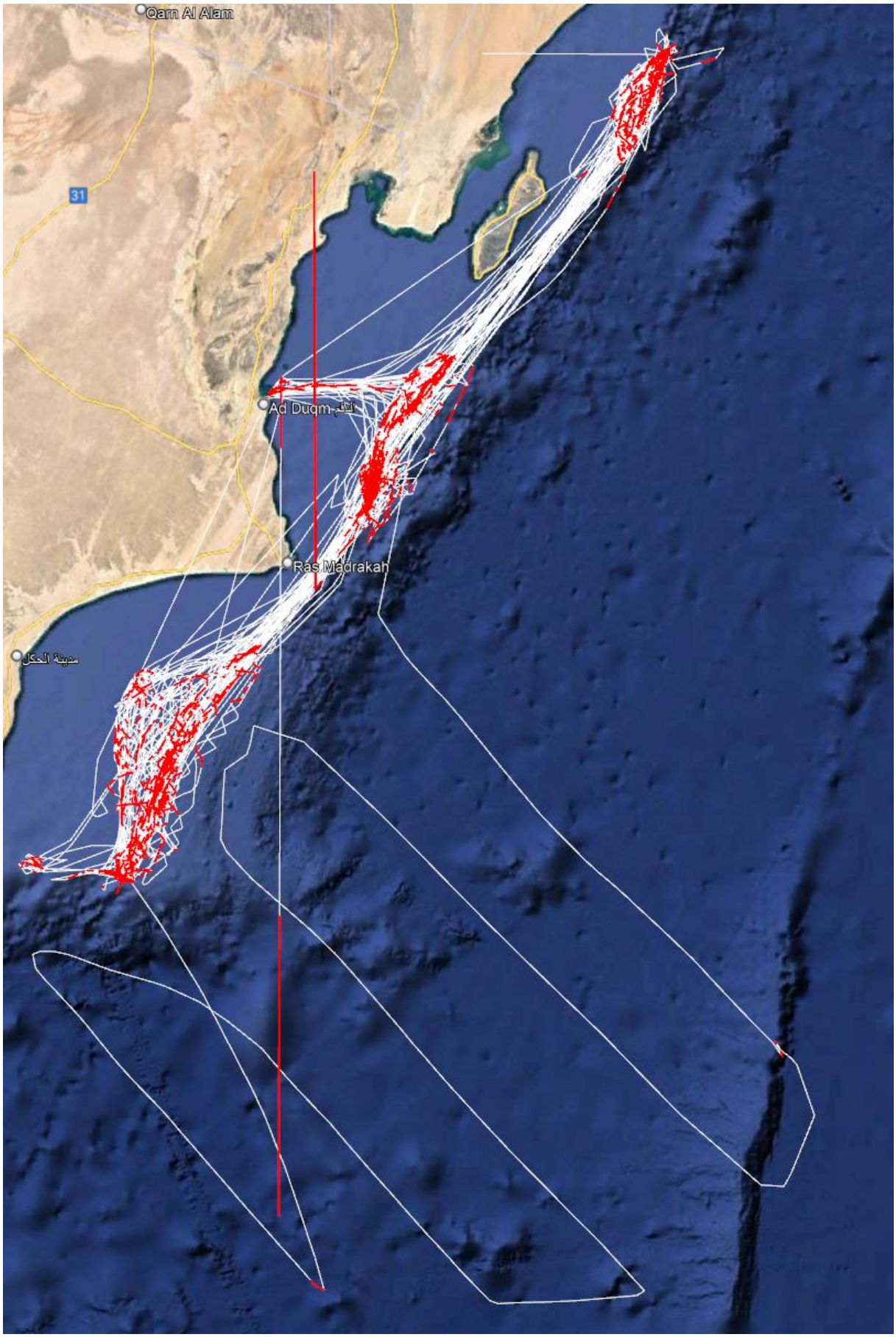


Figure 29: Predicted fishing areas based on model 1 using Google Earth Pro

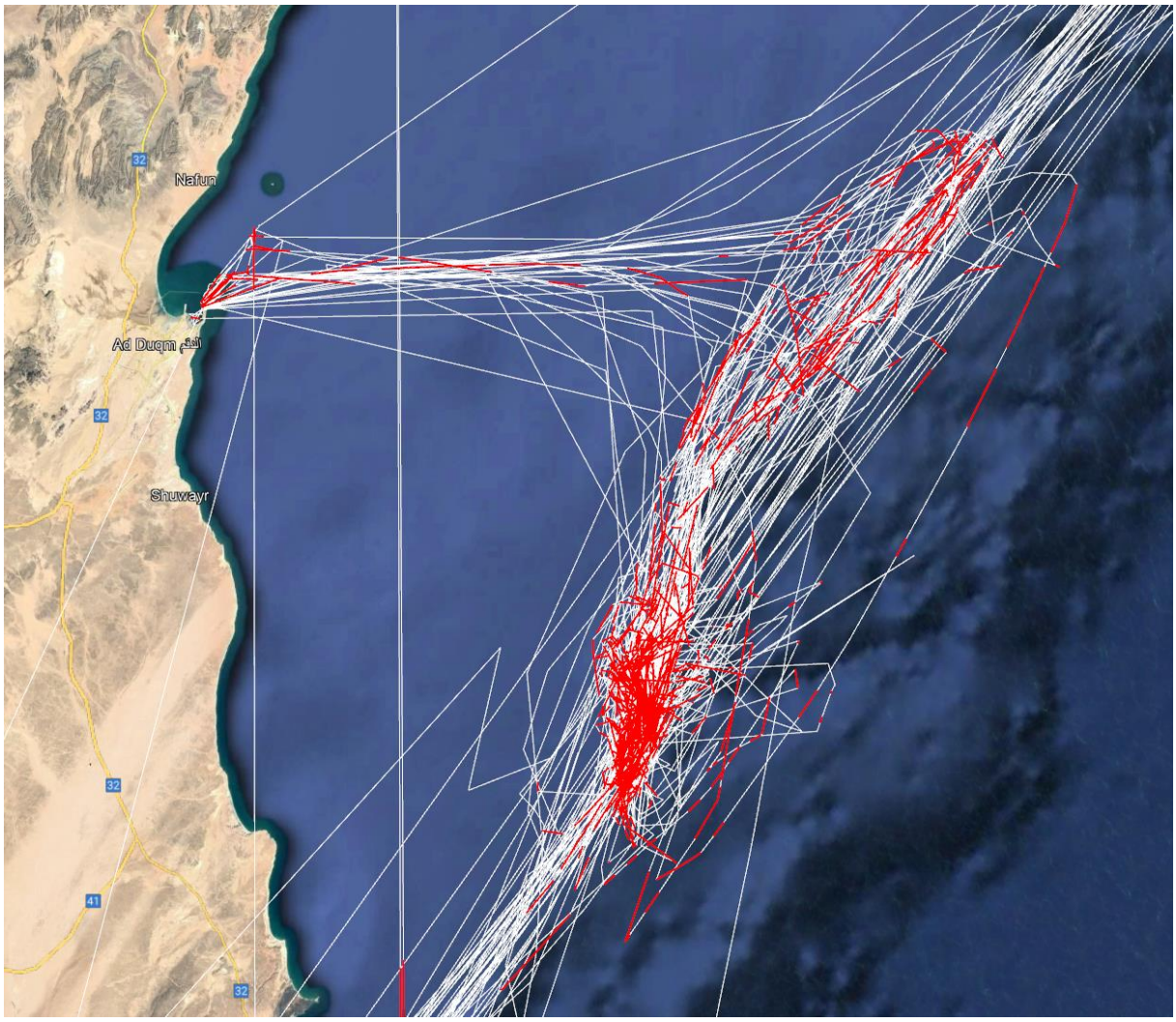


Figure 30: Predicted fishing areas based on model 1 using Google Earth Pro, port area

Appendix 10: Visualization of the validation of model 2

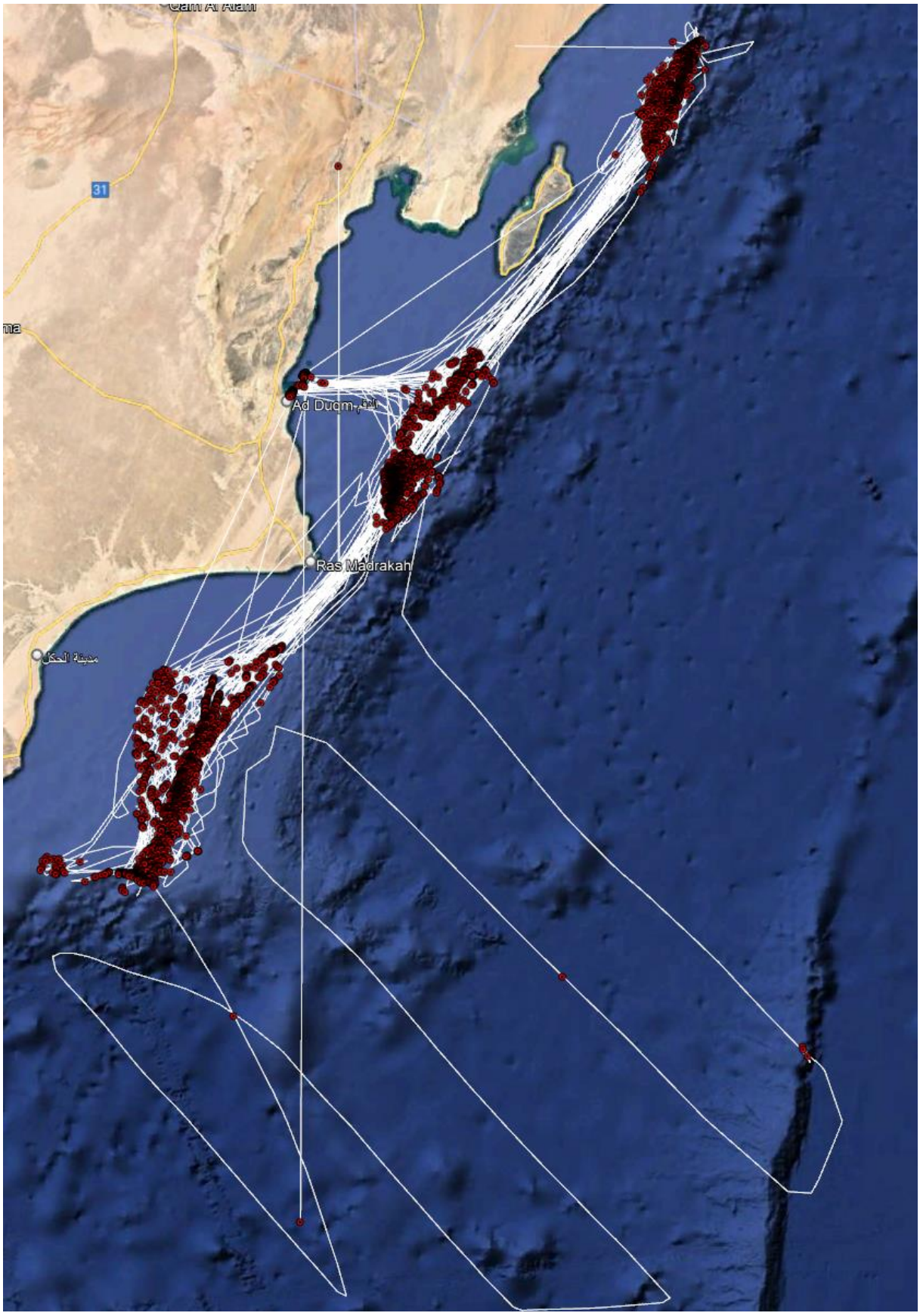


Figure 31: Predicted fishing areas based on model 2 using Google Earth Pro

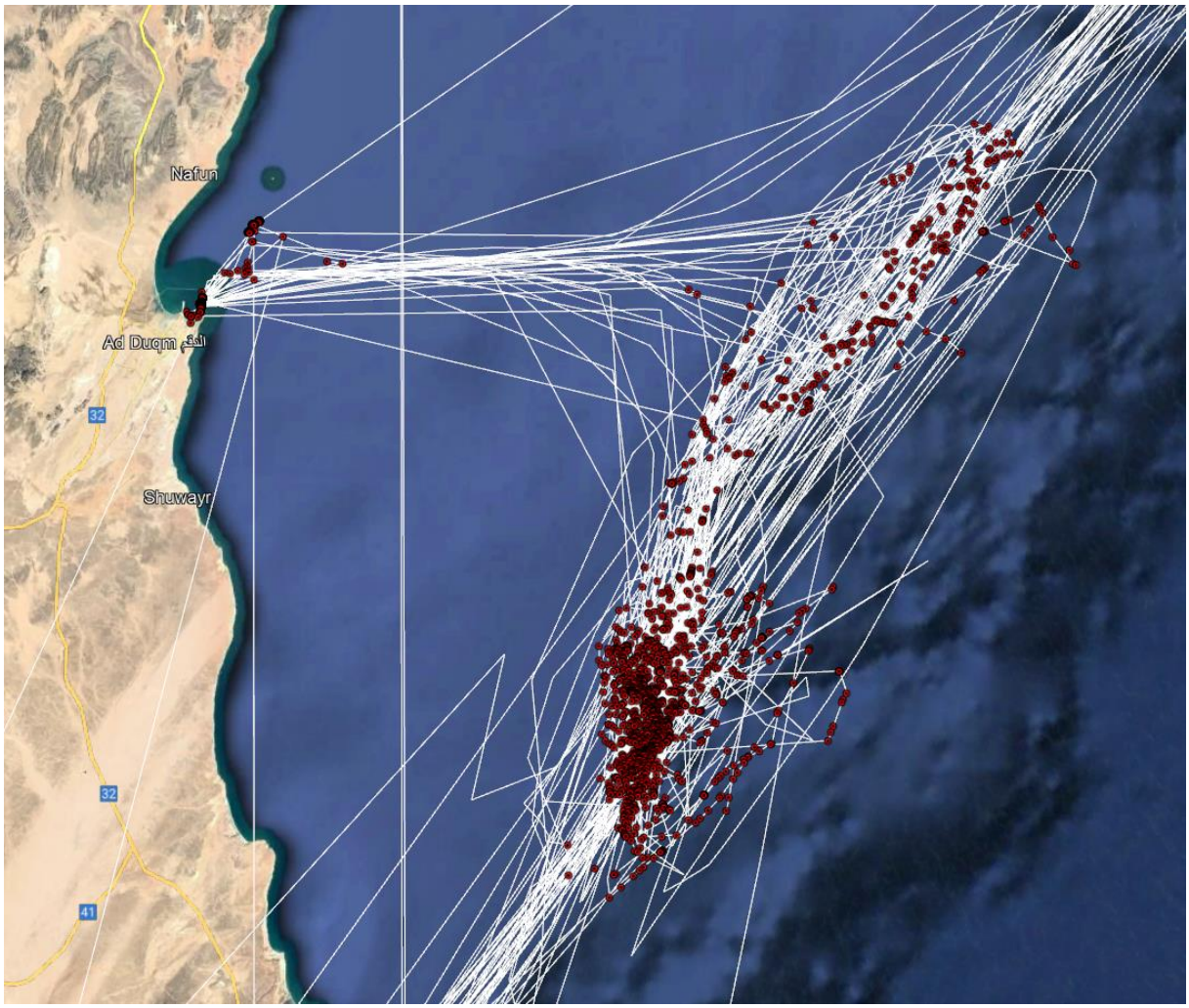


Figure 32: Predicted fishing areas based on model 2 using Google Earth Pro, port area

Appendix 11: Visualization of the Oman area from GFW

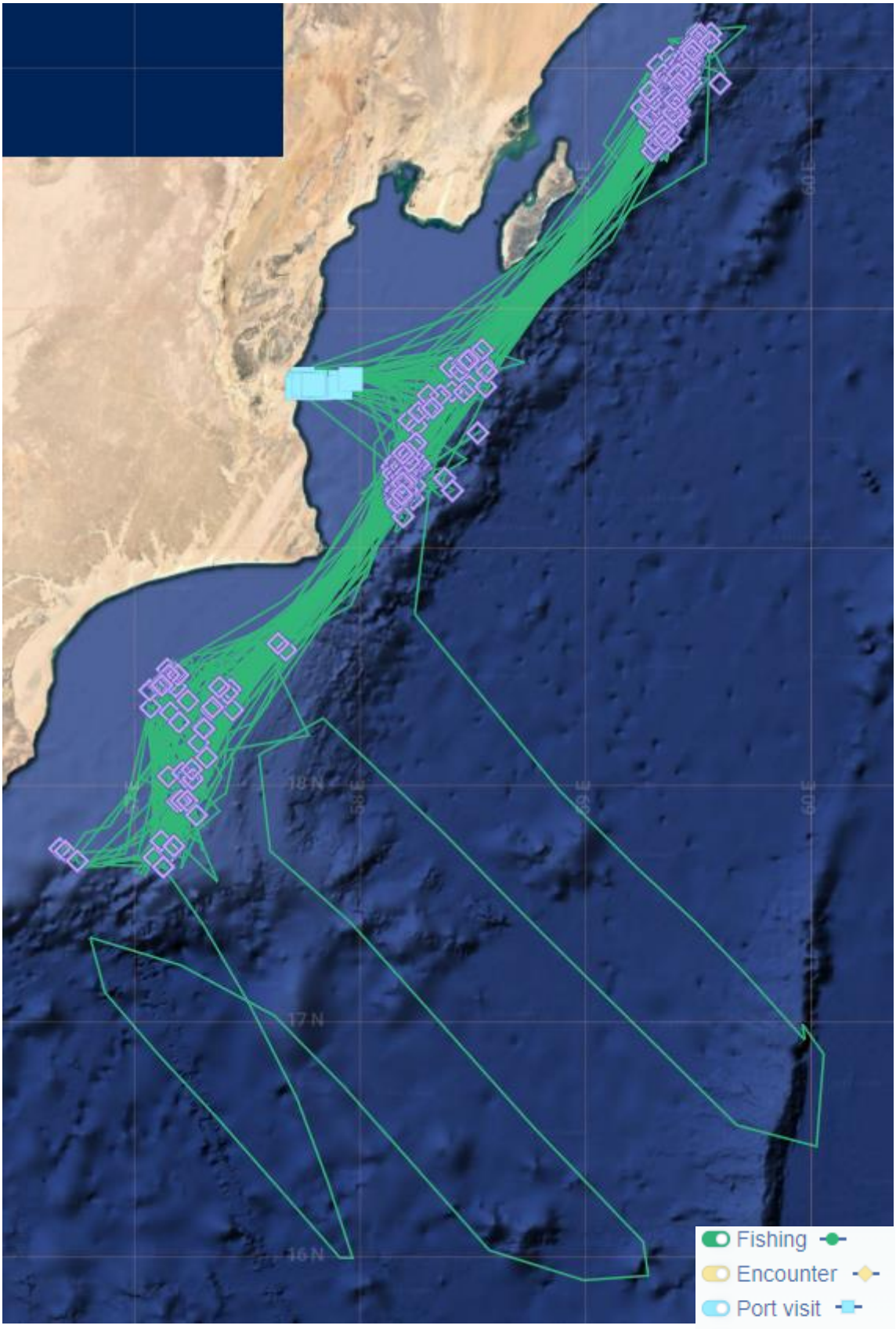


Figure 33: Predicted fishing activities based on GFW

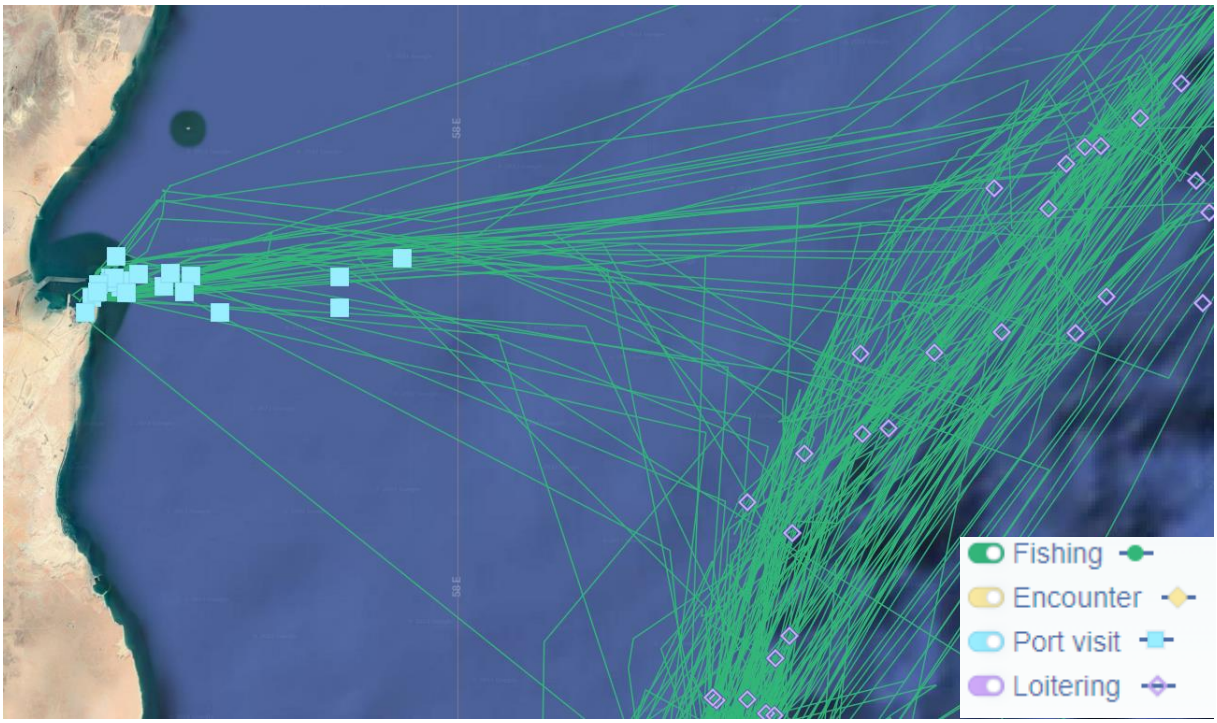


Figure 34: Predicted fishing activities based on GFW, port area

Accessed via:

https://globalfishingwatch.org/map/fishing-activity/from_dec_27_2020_to_jun_10_2022_near_oman-public?dvIn[0][id]=vessel-d78e9e168-8d32-df48-d099-58a5d8748372&dvIn[0][cfg][clr]=%23AD1457&dvIn[1][id]=basemap&dvIn[1][cfg][basemap]=satellite&dvIn[2][id]=presence&dvIn[2][cfg][vis]=false&dvIn[3][id]=vms&dvIn[3][cfg][vis]=false&dvIn[4][id]=fishing-ais&dvIn[4][cfg][vis]=false&dvIn[5][id]=vessel-bb05c04dd-d946-28db-296e-ca15d6849995&dvIn[5][deleted]=true&dvIn[6][id]=vessel-c7f94defb-b8f7-bba4-aac2-dc3bf3fb8f3f&dvIn[6][deleted]=true&dvIn[7][id]=vessel-1c2a3fbdb-b8be-f24c-5c11-071bd9758953&dvIn[7][deleted]=true&dvIn[8][id]=vessel-f6e810b9d-dd52-86ea-1a3c-e0546565df14&dvIn[8][deleted]=true&dvIn[9][id]=vessel-f11696e05-5ffd-3691-445a-20472297ee57&dvIn[9][deleted]=true&latitude=18.30499242329536&longitude=58.827901584827714&zoom=6.631376528715246&start=2020-09-05T00%3A00%3A00.000Z&end=2022-06-10T23%3A59%3A59.999Z&visibleEvents[0]=encounter&visibleEvents[1]=port_visit&visibleEvents[2]=loitering&visibleEvents[3]=fishing