Norwegian School of Economics
Bergen, Spring 2023

Port waiting time for oil tankers

Leveraging AIS data to predict port waiting time using machine learning

Eric C. Glenjen & Kristoffer O. Solberg

Supervisor: Gabriel Fuentes

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

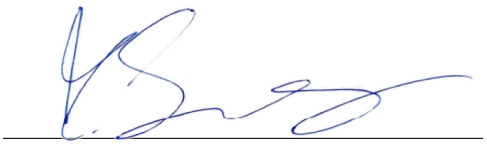
Acknowledgements

In most workplaces there are a handful of people who always deliver a little extra. For an educational institution these people are invaluable, as they serve as role models and inspiration to students in the onset of their careers. We can confidently claim that our supervisor Gabriel Fuentes is one of those people. After only a couple of lessons in the course ENE431 Shipping Economics and Analytics, our interest in data analysis within the maritime domain was sparked. With this we would like to thank Dr. Fuentes for his well-balanced support, spurring creative and critical thinking, as well as excellent practical facilitation.

We would also like to thank the UN Statistics Division for granting us access to the UN Global Platform. This allowed us access to a database of clean data, the tools to handle such vast amounts of data, and the opportunity to concentrate on proper methodological work. Something which has been a luxury. Our sincere wish is that, apart from being the completion of our degree, our efforts in this thesis will prove useful to someone, somewhere.

Norwegian School of Economics

Bergen, May 2023



Kristioffer Overholdt Solberg

Eric Christopher Glenjen

Abstract

This master's thesis investigates the predictability of waiting times at crude oil ports using Automatic Identification System (AIS) data and machine learning. Focusing on the wet bulk market, specifically four congested Middle Eastern Gulf ports, we aimed to answer: "*Can the waiting times in crude oil ports be predicted based on AIS data?*". In this thesis clustering algorithms with novel modifications are utilized to establish berth and anchorage polygons. These polygons form the basis for a spatial matching of AIS data that is used to generate event logs. A cross-sectional data set is derived from the event logs which in turn is the basis for extracting features used in five different machine learning models. The findings show that AIS-derived features have predictive power on waiting times, with vessel composition within ports and port dynamics being significant factors. These insights hold practical implications for ship owners and academics alike, enhancing vessel economics through speed adjustments and facilitating further research within the maritime domain. The thesis also proposes further research areas, including methodology refinement within polygon generation, event log generation and waiting time prediction.

Keywords – Shipping, Tanker, Tramp, AIS, Queue, Machine Learning, Artificial Intelligence, Port Congestion, Clustering Algorithm, DBSCAN, Polygon Generation, Waiting Time Prediction, Big Data, Distributed Computing, Port Operations

Contents

List of Figures

List of Tables

1 Introduction

1.1 The Importance of Data in Decision Making

The situation when a vessel calls at a port is an especially interesting part of the voyage as the complexity of the decision environment often increases. Coordination with other entities such as other vessels, tugs and port authorities, as well as uncertainty around waiting times and service times, complicates decision-making. These decisions involve matters such as maintenance activities while in port, determining if the ship is available for new contracts, or deciding the best time to call at a port to avoid waiting times.

Decisions in modern businesses are increasingly data driven, with the shipping industry being no exception (Munim et al., 2020). In a complex environment, decision makers across all levels in a maritime organization must balance factors such as risk taking, safety, profit maximization and compliance. Having data that sheds light on various aspects of the decision-making process enhances the margin of safety, both in terms of financial risk reduction and physical safety. Leveraging data not only helps stakeholders within a maritime organization, but also external stakeholders such as charterers, insurance companies, authorities and academia. A clearer understanding of port dynamics and vessel behaviour is key to saving resources, abating emissions and creating efficient ports in the future. A concrete example of this is the topic of speed adjustment due to port efficiency, where insight in to an expected waiting time supports a sound decision of vessel speed.

A significant amount of fuel could be saved if vessels optimized speed to avoid arriving at a port in periods of high congestion (Andersson and Ivehammar, 2017). On average, as much as five to ten percent of a vessel's time is spent at anchorage or maneuvering at slow speeds, waiting to get into port (International Maritime Organization, 2020). In short-sea shipping, these numbers are even higher. Despite this, vessels keep arriving at ports earlier than necessary, regardless of berth availability (Johnson and Styhre, 2015). Although sensible in theory, speed adjustment for more timely arrival at ports has proven difficult in practice. The barriers to just-in-time (JIT) arrival span from contractual issues, misalignment in incentives between shipowner and charterer, and lack of data for making

decisions (International Maritime Organization, 2020).

The lack of data poses a tactical barrier to making timely decisions. However, this barrier has been reduced by the increased availability of vessels' spatial data, provided by the Automatic Identification System (AIS). Originally introduced as a safety measure, AIS data is now used in a wide variety of maritime research, bringing some transparency into an otherwise complex and opaque business (Svanberg et al., 2019; Yang et al., 2019). Using machine learning, researchers are now able to leverage AIS data to predict port characteristics, such as efficiency, pollution and congestion, to name a few. As liner services are more predictable than tramp shipping, the majority of research on port efficiency and congestion is focused on liner shipping (Filom et al., 2022; International Maritime Organization, 2020).

1.2 Problem Statement

In this thesis, existing algorithms and machine learning techniques will be used to leverage AIS data to provide insight into a less researched part of the shipping literature, namely congestion in the wet bulk market. The fundamental research question for this thesis is:

"Can the waiting times in crude oil ports be predicted based on AIS data?"

1.3 Contribution

The contribution of this thesis is to add to existing machine learning literature in maritime applications and re-contextualizing existing algorithms in novel ways. In doing so, we will contribute to the topic of congestion in oil terminals. The focus is on the use of AIS data and the automated collection of port characteristics. The methods in this thesis can be adapted to new uses by both practitioners and academics without relying on inaccessible and unreliable data provided by ports or port authorities (Slack et al., 2018).

1.4 Scope

AIS data is reported from a vessel at intervals of every two second to every sixth minute, with each report consisting of 19 variables (International Telecommunication Union, 2014). All vessels above 300 gross tonnage, cargo vessels above 500 gross tonnage and all passenger

vessels irrespective of size, is to be fitted with AIS (International Maritime Organization, 2002). Given the number of vessels fitted with AIS and interval of reporting, AIS data sets can be large and impractical if no proper plan for reducing the data is implemented.

To reduce computational labor and enable some degree of manual cross-checking of data, the ports included in this thesis will be a selection of four ports in the Middle Eastern Gulf, with known congestion. This data reduction allows for expanding the temporal aspect, making it possible to train the models on longer time spans, increasing the possibility of picking up seasonal effects in the data. Our emphasis on crude oil ports is based on the expectation that vessels calling at the port carries a homogeneous cargo, which is assumed to show little variability in port cycle time.

1.5 Outline

The rest of the thesis consists of five sections. First, a literature review of relevant topics touched upon in this thesis (AIS research, port characteristics mining and machine learning for prediction of waiting times). Second, a description of the methodological setup of data processing and machine learning models used for predictions. Third, the results of the machine learning models will be presented, along with a selection of descriptive statistics for each port. Fourth, the key findings will be highlighted in the discussion section along with identified methodological limitations. Finally, we conclude and propose recommendations for further research.

2 Literature Review

2.1 The Automatic Identification System

The International Convention for the Safety of Life at Sea (SOLAS), requires ships to be equipped with AIS (International Mariime Organization, 2019). The main goal of AIS was, and still is, navigational safety, ship reporting and ship-to-ship communications. It relied on VHF signals between base stations placed at the coast and the ships. As a consequence, problems could arise whenever the vessels were too far away from land (Emmens et al., 2021). In later years the use of satellites, in tandem with the coastal base stations, has improved this situation and increased the coverage of the AIS (Emmens et al., 2021). Regardless, the AIS data may lack coverage and may not produce complete information (Silveira et al., 2015; Emmens et al., 2021).

Svanberg et al. (2019) outlines how flexible AIS is as a data source. They find that AIS data is suitable as the sole source of data, or combined with other sources of data e.g., weather data. Additionally, they find that AIS data can be used as a secondary data source in the validation of the primary data source's results. As AIS data is collected in near real-time (International Telecommunication Union, 2014), it can be used to improve the supply chain at various points. This is in line with the literature review by Svanberg et al. (2019), who found that AIS data is playing a more important role in economics and logistics research. Steenari et al. (2022) used AIS-data to mine port operation information, providing a proof-of-concept for an approach to automatically keep mooring areas up to date. Fuentes (2021) utilized a machine learning approach to identify bunkering statistics from AIS data, successfully identifying a high percentage of bunkering operations in ports in the Mediterranean and Marmara seas. Zhang et al. (2018) used AIS data in their construction of a vessel trajectory reconstruction model in a real-case scenario, tackling the noise of raw AIS data.

The use of AIS-data is not without its challenges. AIS data tends to contain a lot of noise such as errors in positioning, speed or timestamps (Emmens et al., 2021; Zhang et al., 2018). Proper filtering of raw AIS data will do a good job of contending with any noise present (Zhang et al., 2018; Dobrkovic et al., 2018). This should also assist

with the overabundance of information often found in AIS data (Emmens et al., 2021). Furthermore, one also has to evaluate the quality of the AIS data at hand which can vary depending on one's source, which can be controlled. The AIS equipment fitted on ships as well as the base stations also contributes to the quality of the AIS data (Emmens et al., 2021; Zhang et al., 2018). Vessels that match the aforementioned specifications of tonnage and voyage types are required to be fitted with class A AIS equipment, which is more expensive and of better quality. Other ships, such as smaller ones, can be fitted with class B AIS equipment, which is cheaper and simpler (Xiao et al., 2015). There are differences in the quality of AIS records based on which geographical area is investigated. For example, Xiao et al. (2015) found that AIS data collected from a Chinese inland waterway was of significantly poorer quality than that of a Dutch inland waterway.

2.2 Mining Port Characteristics

One of the main challenges in port-efficiency studies is that data about port performance are of a commercially sensitive nature. Furthermore, data obtained from industry or port authorities are often low-frequency and different from port to port (Peng et al., 2022). This limits researchers' accessibility and usability of these kinds of data sources. However, automated extraction of port characteristics, such as congestion and vessel cycle time, amongst others, have gained traction as AIS data has become widely available. In a comprehensive literature review, Filom et al. (2022) investigates the applications of machine learning methods in port operations in the current literature. The literature is split into five different applications, demand prediction, land-side operations, seaside operations, safety, and other applications. Most of the studies in this review focus on containerized cargo, indicating that there is room for complementing research focusing on the tramp market.

Seizing the potential opportunities AIS data present, requires the researcher to address how to effectively extract relevant information. Often, elaborate processing is required to fully leverage the potential of AIS data. Several research papers put a sizeable amount of effort into this step. Academic articles analysing port congestion indicators, waiting time in anchorage or vessel cycle time, often have similar initial steps to extract insights. First, anchorage and berth areas are identified, second derived data are generated using

AIS data and the berth and/or anchorage areas, lastly descriptive statistics or machine learning models are used to analyse the derived data.

Millefiori et al. (2016) outlined a method to estimate sea port operational regions from AIS data. Using World Port Index (WPI) data, the authors use k-Nearest Neighbor, a clustering algorithm, to allocate a port to each AIS message in a large data set. In turn, these AIS-port-clusters are used to define logical spatial partitions for each port, representing the operational region of that port. A kernel estimation function is then applied on each region to further narrow the area of interest. An advantage of this approach is the scalability. However, this method only yields an area of operation for a port, with no segmentation between quay or berth areas and anchorages.

In 2018, Abualhaol et al. proposed three methods for mining port congestion indicators: convex hull, geohash area and vessel proximity. The authors derived three congestion indicators, spatial complexity, spatial density and time criticality. Whereas the first two are measures of vessel density in the port area and the last is a measure of average cycle time for vessels in port. Central in both spatial complexity and spatial density is the convex hull algorithm. This algorithm is the smallest convex set that contains a set of points (Barber et al., 1996). Used in the context of AIS data mining, convex hull represents the bounding latitudinal and longitudinal points in a cluster of AIS messages.

While Abualhaol et al. (2018) does not differentiate between anchorage time and berth time when calculating the average cycle time, Peng et al. (2022) does. In their study, they propose high-frequency container port measures, based on AIS data. Berth and anchorage areas are identified for a 200km radius within the top 20 container ports. A combination of Density Based Spatial Clustering of Applications with Noise (DBSCAN) and convex hull is used to establish polygons for berth and anchorage areas. Used on AIS data, DBSCAN clusters AIS messages in the spatial plane, filtering out noise points, only leaving clusters of points within a given distance from each other. The polygons yielded from this process are used to calculate the number of ships in each polygon each hour, the total dead weight tons (dwt) present in each polygon, and the average turnaround time in each polygon. These features are fed in to a recurrent neural network (RNN), yielding either a point prediction or a sequence prediction for estimated waiting time. In addition, the authors factor in a congestion propagation effect, modeling the spillover effect from

congestion in one terminal to other terminals.

DBSCAN is also used as the basis for mining port operation information from AIS data by Steenari et al. (2022). They use the centroid of the AIS clusters produced by each mooring event. There are several advantages to this approach, one is that the presence of stray positional points in each cluster will not significantly affect the final shape of the berth polygons. Also, using only one point, the centroid, from each mooring event, allows for using data from additional mooring events. This allows for the algorithm to capture vessels moored at slightly different positions at the quay, as well as different positions of the AIS transceiver aboard the vessel. In this paper, they also investigate different settings of the DBSCAN hyperparameter epsilon, which refers to the allowed distance from a core point in a cluster. Adjusting this point, yields different clustering results in terms of defining a cluster as the whole quay area or simply a single berth.

2.3 Waiting Time Prediction

Although researchers have developed methods to extract port efficiency metrics, including congestion indicators, there is a lack of literature on predicting waiting times in the tanker sector. It is worth noting that queue analysis and wait time prediction have been more extensively researched in other domains such as traffic, food service, and medical sectors.

Some of the methods used for predicting wait times are: rolling averages, queue theory, discrete-event simulations and machine learning models. Sanit-In and Saikaew (2019) investigates three approaches for predicting waiting time in a one-stop service problem, more specifically in a medical clinic and a post office. The authors compared the predictive performance of queueing theory, average time and random forest machine learning algorithm. Noteworthy of this paper, is that the authors binned the waiting times into five classes, turning the problem in to a classification problem. Although at the expense of precision, this approach requires less training data. Random forest yielded the best results with an accuracy of just above 85% on the medical clinic data set, and just shy of 87% on the post office data set. In both of these, the length of the queue was the most important variable to increase accuracy.

In a study from 2019, Kyritsis and Deriaz showed that a lean machine learning model with only four predictors achieved a significant improvement over using just the naive

mean or the naive median as an estimator for waiting time. Along with queue length, three temporal estimators were used, day of week, hour of day, and minute of hour. These predictors were fed in to a fully connected neural network with two hidden layers, consisting of twelve and eight neurons in the first and second layer respectively. This resulted in an improvement in close to 30% in the mean absolute error, using the median and mean waiting time as benchmark.

In an attempt to predict waiting times at two Italian emergency rooms (ER), Benevento et al. (2021) used a set of predictors, describing the patient and the state of the queue, along with temporal predictors and arrival based predictors. By implementing arrival-based predictors, the authors factored in that patients at an ER are triaged based on their condition and not treated on a first come, first served basis. For predicting waiting times, the authors ran experiments using lasso regression, random forest, support vector regression, artificial neural network and an ensemble model, using the weighted sum of all the other models. The best performing model was the ensemble model with a mean average error of 20.8 and 46 minutes for ER 1 and ER 2 respectively.

3 Methodology

3.1 Methodological Overview

Going from raw AIS data to a prediction requires a fair number of intermediate steps, described in Figure 3.1. The methodological process can be split into three parts:

1. **Establish port and anchor areas.** In this part, the aim is to establish polygons representing anchorage areas and individual berths.
2. **Generate event logs.** Berth events with accompanying waiting times are generated on the basis of the polygons generated in the first step.
3. **Predict based on event logs.** Based on the statistics captured in the event logs, a set of predictors is derived for use in five machine learning models.

Each part is composed of numerous sub-parts described in the sections below.

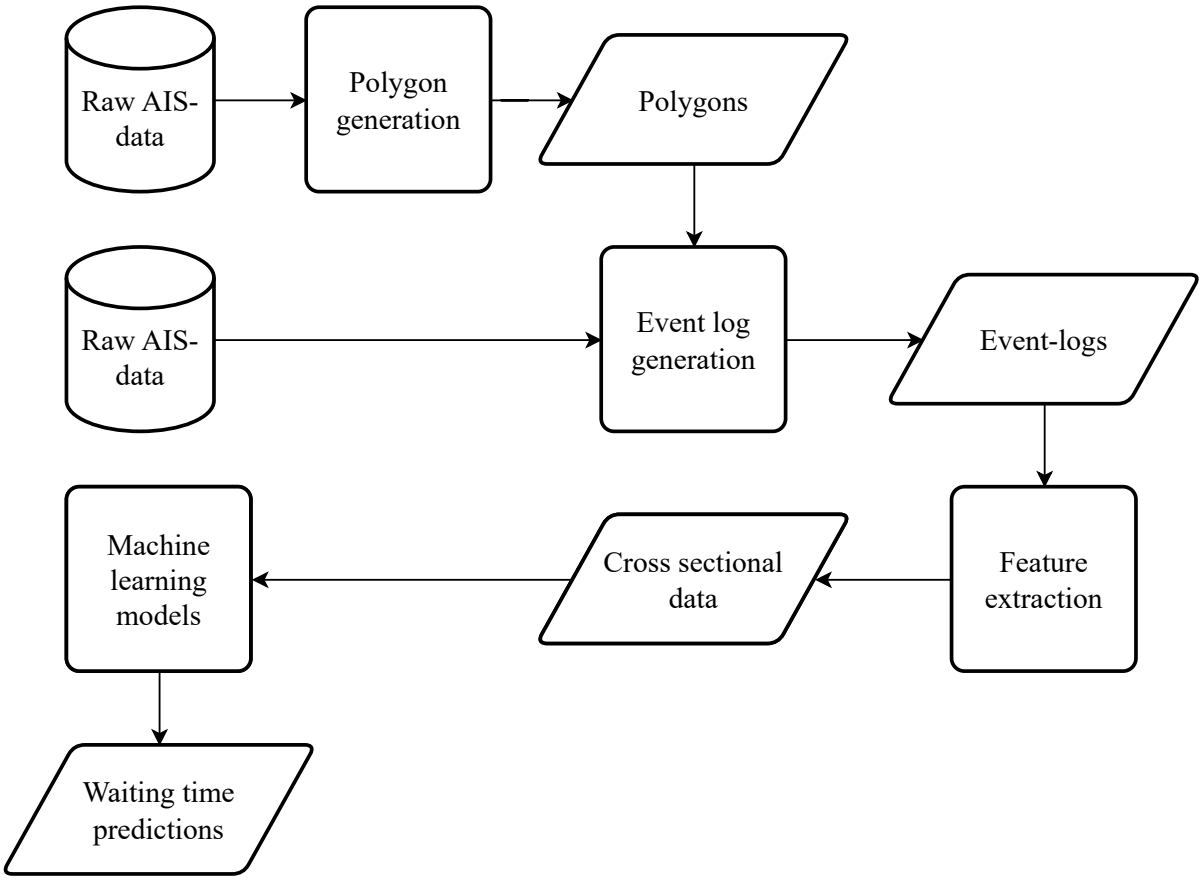


Figure 3.1: Data flow overview

3.2 Big Data

Due to the number of vessels at sea and the frequency of AIS transmissions, the number of observations in an AIS data set quickly get in the tens of billions. Handling all of this data requires a thoughtful data pipeline and a distributed environment to complete the task at hand within a reasonable time frame.

AIS data provided by United Nations Global Platform (UNGP) is extracted from a Spire API, cleaned and stored in an Amazon Web Service (AWS) S3 bucket, which is part of the AWS data lake system. With Apache spark running on top of Kubernetes, users are able to leverage distributed computing and dynamic resource allocation to access and manipulate the AIS data (UNSD, 2023).

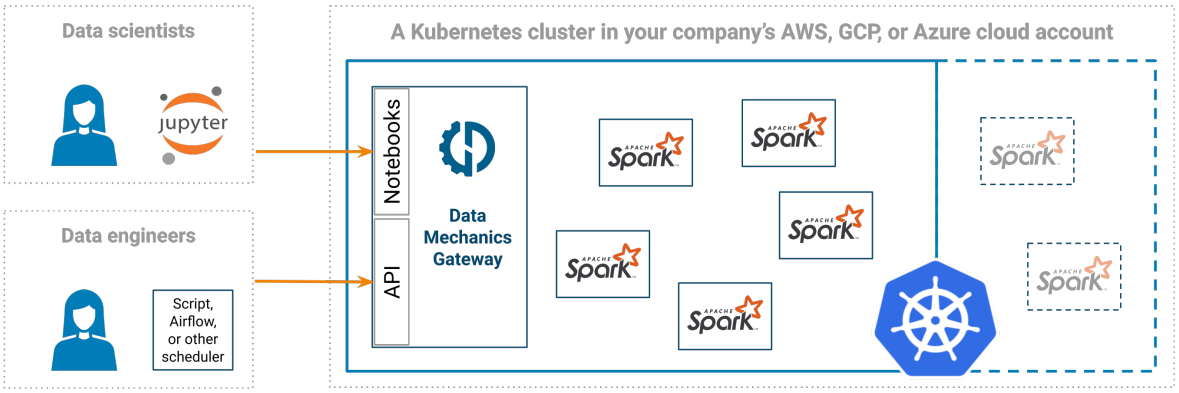


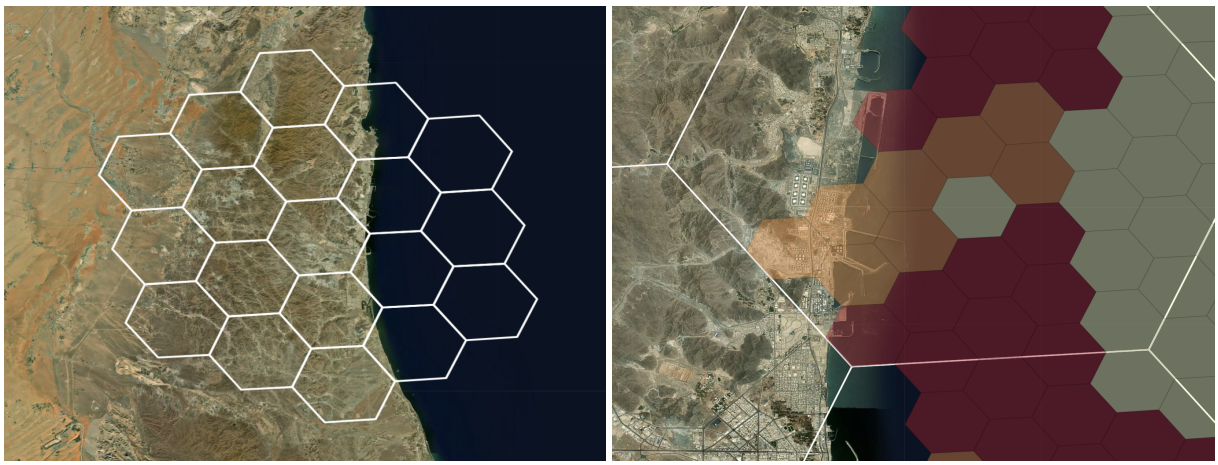
Figure 3.2: The cloud architecture. Image source: Data Mechanics (2022)

3.3 Establishing Port and Anchorage Areas

Being the foundation for generating event logs, establishing precise polygons for berth and anchorage areas, are of utmost importance. The heterogeneous nature and complexity of port areas poses several challenges. Especially when an automated algorithm is expected to distinguish between noise and signal across several different ports. In this context, noise could be false berth positions generated by the algorithm on the basis of ship activity that resembles a mooring event, for instance a vessel mooring for maintenance at a dock or ship to ship transfers. Several measures are taken to mitigate the probability of false polygons.

Port positions are retrieved from the World Port Index database, published by the National

Geospatial-Intelligence Agency (National Geospatial-Intelligence Agency, 2020). These positions form the center of a hexagonal search grid, which has a radius of approximately 40 kilometers. The search grid, shown in Figure 3.3a, defines the area from which AIS data is retrieved. A vessel type filter, based on the UN Ships Register Data, is applied on the AIS data, leaving only AIS data from crude oil tankers. Furthermore, for each vessel every second observation is removed to conserve computational resources. This way, the time frame from which data is retrieved from can be doubled, capturing additional mooring events, as opposed to every single detail for each mooring event. The port area is then partitioned into mooring areas and anchorage areas. Within the mooring areas, AIS messages are clustered to establish the exact berth positions. Three rounds of clustering are performed, the first to differentiate between traditional berth positions and single point moorings (SPM), the second is to establish berth position clusters and lastly SPM clusters are established. Anchorage areas are defined as areas containing AIS messages with navigational status *At Anchor* and speed above ground below five knots.



(a) Search grid **(b)** Partitioning of port area

Figure 3.3: Fujairah port in the United Arab Emirates

3.3.1 Partitioning the port area

Port area definition and partition is done using the Python API of the H3 library. This library is an implementation of the H3 Hexagonal Hierarchical Spatial Index, which is a discrete global grid system based on hexagonal cells. H3 has 16 cell resolutions, meaning the system covers the earth using 122 of the cells with resolution 0, and well above 500 trillion cells of the finest resolution (15). Each cell in the H3 system has its own unique

string and integer identifier, making big data handling efficient. The hexagonal shape allows for covering a spherical object such as the earth with hexagons of the same size. An advantage of using hexagons is that the distance from the center of one cell to all neighboring cells are equidistant. The different resolutions also provide flexibility in use.

When a vessel moores at a berth, the navigational AIS status is manually set to *Moored*. However, this navigational status is also sometimes used when conducting ship to ship transfers, when laid up in dock for maintenance or inadvertently at the wrong time or place. For the purpose of identifying permanent berth positions of a port, these AIS messages are considered noise. To avoid false berth clusters being formed outside the actual mooring areas, the area within the search grid for each port is divided in to hexagons of resolution 7 (see Figure 3.3b). For each hexagon the navigational status of each AIS message is counted, and the modal category of the navigational status is assigned to each hexagon. By using the modal value for each cell, errors due to wrong navigational status input from some vessels are minimized. This process partitions the port area in to zones based on the most common navigational status observed in each hexagon. The red area represents areas where *Under Way Using Engine* navigational status is the most common, the beige represents *At Anchor* and the orange *Moored*.

3.3.2 Clustering

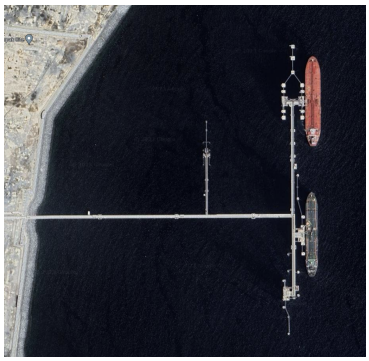
The shape of the berth polygons depends on the mooring arrangement of the specific berth. For this thesis, three types of mooring arrangements are of special interest, due to the distinct shape created by the AIS message positions.

1. **Piers** - Land based structures running parallel or perpendicular to the shoreline. Dense clusters with large inter-cluster separation is typically formed from this mooring arrangement.

2. **Sea islands** - Piers with no connection to shore. Dense clusters with small inter-cluster separation is typically formed from this mooring arrangement, as vessels are moored paralell with the same heading with the pier in between.

3. **Single point mooring (SPM)** - An offshore structure, typically a buoy. Vessels served by this arrangement is moored to the structure at the bow, with a tug vessel at the aft to control movements. Typically forms large dispersed clusters in a circular

pattern.



(a) VLCC pier - Fujairah

(b) Sea Island - Ras Tanura

(c) SPM - Ras Tanura

Figure 3.4: Crude oil tanker mooring arrangements. Image source: Google (2023a,b,c)

3.3.2.1 DBSCAN

Once the port area is partitioned, a clustering algorithm is used to separate mooring arrangements and identify individual berth positions within the mooring areas. For this purpose, an algorithm capable of producing clusters of arbitrary shape with a minimum prior domain knowledge of input parameters is needed. The same needs were also identified by Ester et al. (1996), who developed the DBSCAN algorithm. DBSCAN groups points based on the proximity of the points to other points and the number of other points in the neighborhood. Two hyperparameters are needed, min points (MinPts), which refers to number of other points in the neighborhood and epsilon (Eps), which refers to the size of the neighborhood. The algorithm starts by classifying an arbitrary point. If the point has at least MinPts points within a distance Eps, the point is classified as a core point. However, if the point has less than MinPts points within its neighborhood, it is classified as a border point. A point is classified as noise if it has no points within Eps distance.

The DBSCAN algorithm runs over a distance matrix, which contains all inter-point distances. Most commonly used is euclidean distance. However, in the case of spatial clustering, the euclidean distance metric does not factor in the earths curvature. For small distances, the inaccuracy deriving from a euclidean metric is small. But, the accuracy is easily improved using haversine distance, albeit a bit more computationally demanding. The haversine distance is the distance between two points across a spherical surface. Formula 3.1 shows the calculation of distance d, between two points on a sphere.

$$d_{ij} = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_j - \varphi_i}{2}\right) + \cos\varphi_i \cdot \cos\varphi_j \cdot \sin^2\left(\frac{\lambda_j - \lambda_i}{2}\right)}\right) \tag{3.1}$$

Where:

d_{ij} = Haversine distance from point i to point j

r = Average radius of the earth (6371 kilometers)

φ = Latitude in radians

λ = Longitude in radians

Berth arrangement separation

Although flexible, the DBSCAN algorithm struggles with clusters of varying density, as the optimal hyperparameters are closely related to cluster density. A practical example of this can be seen in Figure 3.5 where DBSCAN has been used to identify pier and SPM clusters. Evident from the figure, the density difference between the pier clusters and the SPM clusters is so large that setting the optimal Eps becomes a trade off between establishing pier or SPM clusters. Figure 3.5a shows that an Eps of 100 meters yields separated berth clusters along the pier, but the SPM cluster in the top right corner is fragmented.

The SPM clusters are formed as vessels are moored to the SPM at the bow, and the AIS transceiver is traditionally located at the bridge. When the vessel pivots around the buoy a circular pattern emerges from the AIS messages. However, an Eps of 100 meters is not large enough to establish a full circular polygon around an SPM when the AIS messages are too sparse. On the other hand, if the Eps parameter is increased, as seen in Figure 3.5b, a full circle around the SPM is formed, but this also results in merged berth clusters along the pier.

The solution to this problem is to cluster pier and sea island berth positions separately from SPMs. SPMs are separated from other mooring points by first clustering all AIS messages within the previously defined mooring partitions of the search grid. For this DBSCAN is used with an Eps parameter of 250 meters and MinPts of 15. This ensures that full circles are formed around the SPMs. The resulting SPM polygons are then

(a) Eps = 100 **(b)** Eps = 250

Figure 3.5: Fujairah port in United Arab Emirates

separated from the other berth polygons based on the area within the polygon and a shape factor. Where the latter is an expression of the area to perimeter ratio. When taking into account that circles have the highest area to perimeter ratio, the shape factor is a good measure of roundness. A perfect circle will have the score of 1, and squares around 0.78. The shape factor is calculated using formula 3.2.

$$Shapefactor = \frac{4 \cdot \pi \cdot Area}{Perimeter^2} \tag{3.2}$$

Where:

Area = The total area of the polygon

Perimeter = The length of the edge of polygon

Figure 3.6 shows the area and the shape factor for all clusters identified within the search grid in Fujairah, Ras Tanura, Al Basra and Al Jubail for the time period 01-06-2022 to 31-12-2022. It is apparent that using the variables area and perimeter allows for classifying a given cluster as either a SPM polygon or a land based/sea island based berth polygon. In this thesis a threshold of 0.00003 for area and 0.8 for shape factor is used for separation.

Once SPMs are identified, hexagons intersecting with the SPM clusters are classified as SPM areas, further partitioning the port area. The SPM areas and berth areas are then clustered separately, resulting in more homogeneous cluster densities.

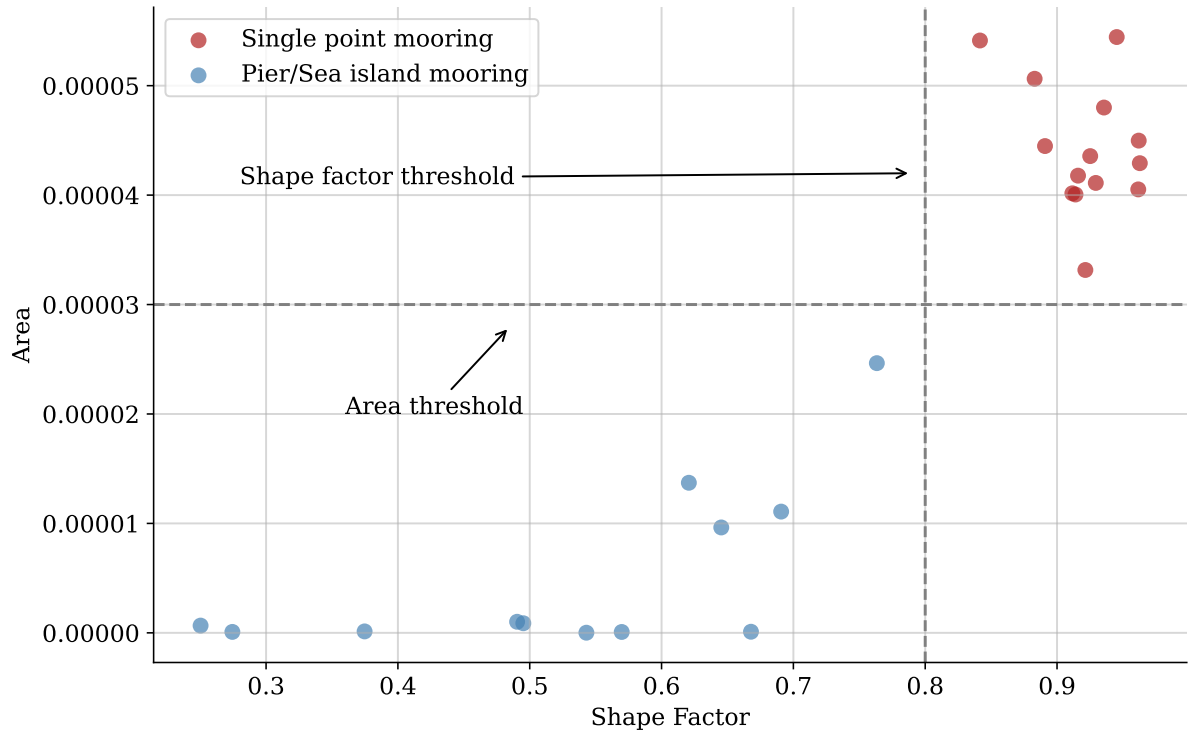


Figure 3.6: Area and shape factor for clusters in Fujairah, Ras Tanura, Al Basra and Al Jubail

Clustering identified berths

In the case of clustering AIS messages with an unsupervised algorithm to identify berth positions, no true cluster label is known. Therefore model performance must be evaluated using other metrics than the traditional accuracy metrics used in classification. In this thesis, the DBSCAN performance is evaluated using the mean silhouette coefficient. The silhouette coefficient is a measure of how well defined each cluster is, and the inter-cluster dispersion. Ranging from -1 to 1, a score close to 1 indicates dense, well defined clusters, a score around 0 indicates overlapping clusters and -1 indicates misclassified clusters. A silhouette coefficient is calculated for each cluster and the mean silhouette coefficient is the metric from which the model is evaluated. The silhouette coefficient for each point is calculated as shown in Equation 3.3 (Rousseeuw, 1987).

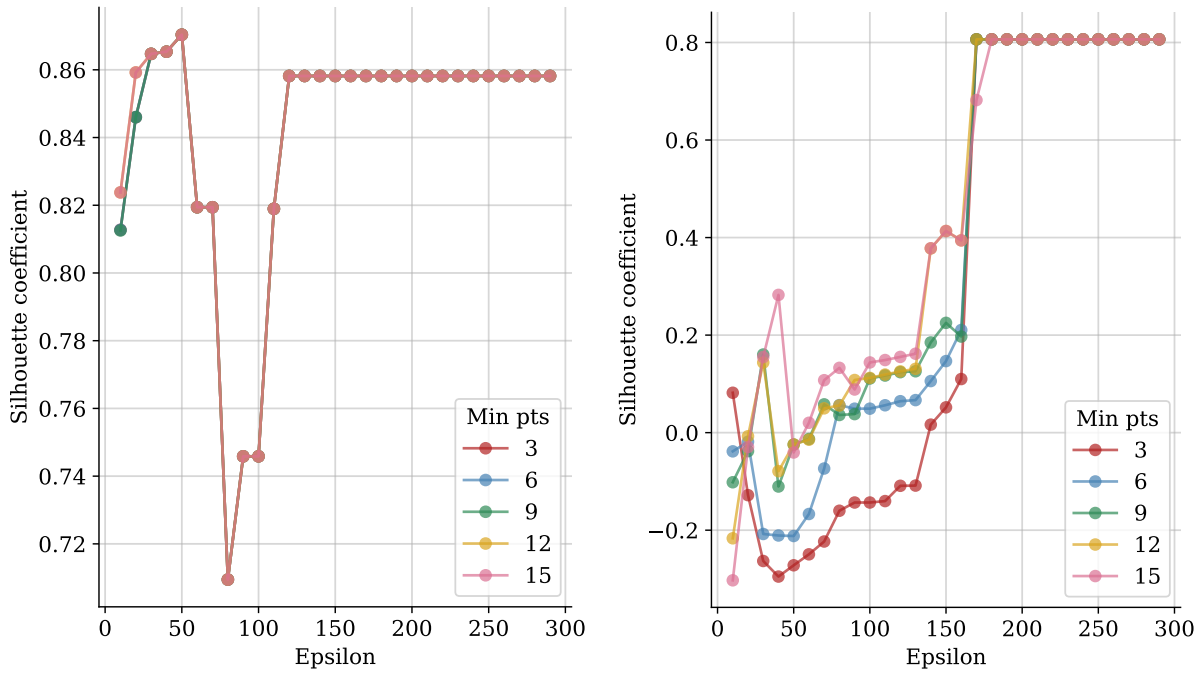
$$s_i = \frac{(b_i - a_i)}{\max(b_i - a_i)} \tag{3.3}$$

Where:

a_i = The mean distance from point i to all other points in the same cluster

b_i = The mean distance from point i to all other points in the next nearest cluster

DBSCAN is run separately for berth and SPM areas of the port, where the hyperparameters are tuned with respect to the optimal silhouette score. The tuning is performed using a search grid with combinations of MinPts (3-18 with step of 3) and Eps (10-300 with step of 10).



(a) Sea Island **(b)** Single point mooring

Figure 3.7: DBSCAN tuning for Ras Tanura

As seen in figure 3.7a, the silhouette coefficient reaches a maxima at a Eps value of 50 meters, before a decrease to a minima of 0.71 at an Eps of 80 meters. The reason for this is that the mean distance from point i to all other points in the same cluster (a_i) increases, as different berth clusters get intertwined with a higher Eps. Thus decreasing the numerator in Equation 3.3. As Eps continues to increase, this effect is partly offset by an increase in b_i as the mean distance from point i to all other points in the next nearest cluster increases. The explanation for this is that the plot depicts the tuning curve for Ras Tanura, which has a sea island of six berth positions. Two and two berth positions are adjacent to each other as seen in Figure 3.7a. As Eps increases, two and two berth positions is considered one cluster. A further increase leads to the entire sea island being considered one cluster, which explains the flattening of the curve at a Eps of 120 meters.

The same process is repeated for the single point mooring areas. In these areas, a opposite effect is seen as fractured clusters (3.5a) closes with an increase in Eps. This process is repeated separately for each port, such that the hyperparameters are tuned to each ports distinct characteristics.

3.3.3 Convex hull

Each cluster generated by the DBSCAN algorithm consists of a set of points. For the purpose of identifying berths or anchorages, only the outer boundaries of the clusters are of interest. Drawing a line through all vertices of the outer points of a cluster to create a convex shape, yields the convex hull of that cluster (Peng et al., 2022). Barber et al. (1996) defined the convex hull of a set of points as *"the smallest convex set that contains the points"*. In this thesis, the convex hull is calculated using the Quickhull algorithm proposed by Barber et al. in 1996. The Quickhull algorithm is a recursive algorithm finding the convex hull of a set of points, and runs until the convex hull for all points are found. A four step method is used:

1. Identify the extreme points along the x-axis, point A and B.
2. Connect point A and B with a line, partitioning the set of points.
3. Identify the farthest point, point C, perpendicular to the line. Point A-B-C represents the vertices of a triangle. Any point within the triangle are disregarded as these can not be part of the convex hull.
4. Find the points farthest away from the sides in the triangle, A-C, B-C, A-B, forming three new triangles. Any points within the triangles are disregarded as these can not be part of the convex hull.

Step 3 and 4 is repeated until no new triangles can be formed. The remaining points are considered the vertices of the convex hull of the set of points. As seen in Figure 3.8.

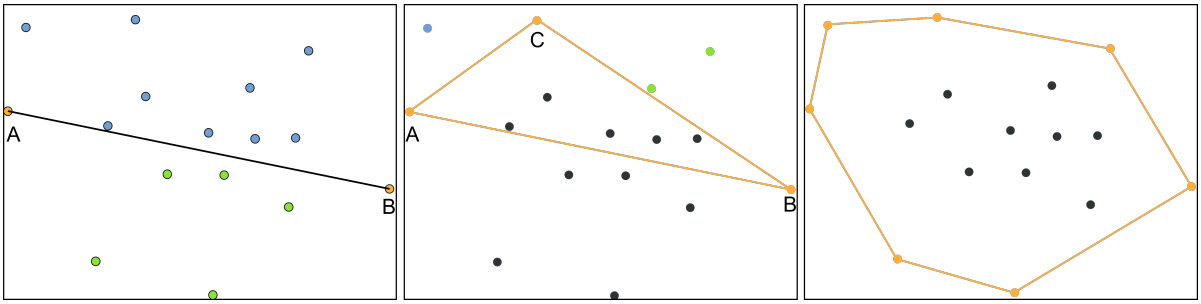


Figure 3.8: Quickhull algorithm. Image source: Wikipedia (2023)

3.4 Event Log Generation

3.4.1 Input

The output of polygons with classifications, H3 integer indices and IMO numbers is used as input in the event log generation. H3 indices enable an inner join on the input and the raw AIS data from the UNGP database. This results in a subset of the raw AIS data that contains only ships in the input polygons. Due to memory constraints on the cluster, this procedure is repeated once for each type of polygon (berth, SPM and anchorage). To avoid generating an unnecessarily large amount of data, an appropriate resolution of H3 hexagons is chosen for each polygon type.

3.4.2 Defining events

A set of events is defined, which each ship must undertake for a port call to be completed. Equation 3.4 makes up a port call to a given port for a given ship. Travel time between anchorages and berths/SPMs will not be explicitly taken into consideration. Rather, it is implicitly included in the calculation of V_n, as one could define it as $S^{\alpha}_{B_i} - S^{\omega}_{A_i}$.

$$V_n = \Delta_{A_i} + \Delta_{B_i} \tag{3.4}$$

$$\Delta_{A_i} = S^{\alpha}_{A_i} - S^{\omega}_{A_i} \tag{3.5}$$

$$\Delta_{B_i} = S^{\alpha}_{B_i} - S^{\omega}_{B_i} \tag{3.6}$$

Where:

V_n = Port call n

S_{ji}^t = Ping from AIS of ship i in polygon j for timestamp t.

Δ_{ji} = Time spent in polygon j for ship i.

j = {A,B} A=Anchor polygon, B = Berth polygon (this also includes SPMs).

$t = \{\alpha, \omega\}$ α = start timestamp in polygon, ω = end timestamp in polygon.

i = Set of unique IMO-numbers.

The key to generating the event log lies in the creation of the start (α) and end (ω) timestamps for each visit on a per-ship basis. To achieve this, all observations with a speed over ground (SOG) greater than 5 knots were removed. This is done to retain only moored vessels. The IMO-numbers in the data are used to facilitate manipulation on a per-ship basis and the timestamps in the AIS messages are used to ensure an ascending order.

To keep track of vessels, change flags are utilized. There are three changes of interest that must be kept track of. (1) A Change in polygon type. Such a change signals that the vessel has moved from e.g. an anchorage to a berth. (2) A Change in Polygon number. This is similar to the previous change, but it allows for tracking across multiple polygons of the same type (e.g. Anchorage 1 and Anchorage 15). (3) A change of trips, i.e. whenever there is more than 48 hours between timestamps in the same polygon. This is important because it allows for separating trips to the same polygon over time. This is critical to obtaining accurate waiting times, as without this, ships observed in a polygon for the first time in 2021, and the last time in 2022, will have an artificially elongated waiting time.

After the change flags are in place, a cumulative sum of these are calculated for each vessel. This is done to separate individual trips, and is necessary to distinguish between trips where all other variables are equal. This can happen whenever a ship visits the same polygon (i.e. berth 1) more than once over time. Individual observations in polygons for each vessel are now identified. In the resulting data, the first timestamp in each polygon, for each vessel, is classified as α and the last as ω.

3.4.3 Output

After identifying the α and ω timestamps, the data is further enriched by classifying vessels. The lack of an official classification led to the use of the McKinsey & Company (McKinsey & Company, 2023) classifications shown in Table 3.1. As one might expect, a lot of vessel types which are used by the ports were included in the data, such as tugs, diving support vessels and bunker barges. Because of this another filtration is performed, this time on vessel types as classified by the Ship Type Coding System (STCS). This is done to retain the types of vessels deemed relevant to this thesis (i.e. tankers). It also helped to remove some noise, and to ensure that the event logs for each port were more uniform. The time spent in each polygon is then calculated for each observation. This information is used to generate statistics and remove noise.

Vessel Class	**DWT Range**
ULCC	320,000–560,000
VLCC	200,000–320,000
Suezmax	120,000–180,000
Aframax	80,000–120,000
Panamax	60,000–80,000
Handy	20,000–55,000
Sub Handysize	Under 20,000

Table 3.1: Vessel classes as defined by McKinsey & Company (2023)

The final step in generating the event logs is to enumerate the trips. Enumeration of trips is performed mainly based on the definition of a port call shown in Equation 3.4. This means that an observation is given a trip number if it is in a berth or an SPM polygon. For an observation in an anchorage to be included, it must be directly followed by an observation in either an SPM or a berth. The implementation of the 48 hour change flags ensured that anchorage observations of ships were not included unless there was less than 48 hours between the last observation in anchorage and the first in berth. An understanding of what observations make up the event log is necessary to gain a deeper understanding of what goes into the prediction models that are tested in this thesis. As such, descriptive statistics from the event log are presented in the results section.

3.4.4 Validation

Due to lack of non-AIS based data for the ports in the Middle Eastern Gulf, the method for extracting port data is validated using data from the Norwegian Coastal Administration (NCA). All vessels above 300 gross tons have an obligation to report port calls in Norwegian waters to the authorities through the SafeSeaNet portal (Kystverket, 2023). The data collected from these reports are publicly available for download at the NCA website. The data is trip-centric, meaning that each trip has its own unique identifier, each port call consists of two observations, the trip to the port in question and the trip from. The data are manipulated in such a way that it is comparable to the event logs extracted in this thesis.

For the validation, port call data from the Norwegian oil terminals Mongstad and Stureterminalen for the period of 2021 and 2022 are used. The data is filtered on vessel type "Crude Oil Tanker", for both the event log and the NCA data. A comparison of number of port calls and time in berth is made. Time in anchorage is not evaluated, as congestion is not common in these ports. The results of the validation is presented in the results section.

3.5 Prediction Based on Event Logs

The event log forms the basis for extracting features which is presented Table 3.2. These features describe various aspects about the vessel calling at the port, the port itself and the queue. A event-centric approach is taken, such that each row in the data-set fed into the machine learning models are a snapshot of a mooring event, with various attributes.

3.5.1 Response variable

Waiting time is defined as the time a vessel spends in anchorage, waiting to get served at a berth. The waiting time starts once a vessels enters a anchorage polygon, and stops once a vessel exits the anchorage polygon, as shown in Equation 3.5. Only vessels that complete a port call, as defined by Equation 3.4, are considered.

3.5.2 Extracted features

Even though many ports practice a first come, first serve queue system, our assumption is that this not always holds true across vessel classes. The reason behind this could be physical limitations for each berth or that certain vessel classes are given precedence for various reason. The vessel related category is supposed to capture attributes about the waiting vessel, that could affect its place in the queue. Furthermore, temporal variables are included to capture seasonality and variations in port staffing trough out the week and day. By capturing the state of each berth using berth-based features, we can determine the occupancy status of each berth as well as track the progress of the current cargo loading or unloading process. In a similar way, the state of the queue is captured through the queue based variables. To account for propagation effects in the queue, lagged variables regarding the previous queue status are also included. One noteworthy feature is the predicted berth which is assigned to each vessel using a random forest classifier. The prediction is based off the deadweight of the vessel mooring. The average waiting time varies based on which berth the vessel calls to. However, from an external point of view, information about which berth a vessel is assigned is not available in advance of a mooring event.

Category	Name	Type
Vessel related	Vessel type	Categorical
	Vessel class	Categorical
	Vessel dwt	Continuous
	Predicted berth	Continuous
Temporal	Hour of day	Categorical
	Day of week	Categorical
	Month of year	Categorical
Berth based	Class of vessel i in berth j	Categorical
	Vessel i dwt in berth j	Continuous
	Time since vessel i moored at berth j	Continuous
Queue based	Total vessels in anchorage	Discrete
	Total dwt in anchorage	Continuous
	Number of VLCC in anchorage	Discrete
	Number of Suezmax in anchorage	Discrete
	Number of Aframax in anchorage	Discrete
	Number of sub-Aframax vessels in anchorage	Discrete
(6, 12, 24, 48 hours)	Lagged total vessels in anchorage	Discrete
(6, 12, 24, 48 hours)	Lagged total dwt in anchorage	Continuous
(6, 12, 24, 48 hours)	Lagged number of VLCC in anchorage	Discrete
(6, 12, 24, 48 hours)	Lagged number of Suezmax in anchorage	Discrete
(6, 12, 24, 48 hours)	Lagged number of Aframax in anchorage	Discrete
(6, 12, 24, 48 hours)	Lagged number of sub-Aframax vessels in anchorage	Discrete
(3, 6, 12, 24, 48, 72 hours)	Lagged mean waiting time	Continuous

Table 3.2: Features extracted from event log

3.5.3 Data exploration and feature engineering

To ensure consistent treatment of the data fed into the machine learning models, a data pipeline is built using the Scikit learn preprocessing module. In the preprocessing step categorical features are encoded, outliers removed and numerical features scaled.

All models, except the feed forward neural network are implemented using the Scikit learn library in Python. Pytorch is used for the implementation of the neural network. The models are trained on a laptop with the specifications presented in Table 3.3. The Scikit learn models are trained on the CPU using 14 out of 16 (8 physical and 8 virtual) cores, and the Pytorch neural network is trained on the NVIDIA GPU.

Component	Specification
Processor	11th Gen Intel(R) Core(TM) i9-11900H @ 2.50GHz 2.50 GHz
RAM	32 GB 3200 Mhz
GPU	NVIDIA GeForce RTX 3070 Laptop GPU

Table 3.3: System Specifications

3.5.3.1 Variable encoding

There are several categorical features amongst the predictors. These are encoded in one of two ways. The categorical features where there is no ordinality amongst the categories is encoded using one hot encoding, meaning each category gets its own column with binary values. Whilst the variable *vessel type*, *vessel class* and *class of vessel i in berth j* is assigned a numerical value from 1 to 7 representing the oil tanker classes in Table 3.1.

3.5.3.2 Scaling

Several of the machine learning algorithms used in this thesis are sensitive to features with different scales, such as deadweight and number of vessels in anchorage. If this difference is not addressed, adverse effects such as feature dominance or inefficient convergence of gradient decent based algorithms could occur. Therefore all numeric features are scaled using Z-score normalization, see Equation 3.7. This sets the mean to 0 and the standard deviation to 1.

$$z = \frac{x - \bar{x}}{\sigma} \tag{3.7}$$

3.5.3.3 Outliers

Only the features concerning waiting time contained outliers that needed to be addressed. The origin of these outliers are likely to be vessels that stay in the anchorage for an extended period of time for different reasons, such as floating storage or waiting for new charter contracts. Also, a glitch in the algorithm assigning each visit to the anchorage a trip number, is a possible explanation. As some of these outliers are extreme, and the data is non-normally distributed, the interquartile range method (Equation 3.8 and 3.9) is chosen over the standard deviation method. Both the dependent variable and the independent wait time features are filtered with this method.

$$\text{IQR} = Q_3 - Q_1 \tag{3.8}$$

$$\text{Outliers} = [Q_1 - 1.5 \times \text{IQR}, Q_3 + 1.5 \times \text{IQR}] \tag{3.9}$$

3.5.3.4 Feature selection

To maintain comparability across models, no feature selection is performed in advance. This implies that every model is fed all features, depending on the number of berths in the port, from 85 to 111 features.

3.5.4 Prediction models

Consolidating the time series AIS-data to cross-sectional data simplifies the implementation of numerous of machine learning models. In this thesis, we aim to compare the predictive power of five different machine learning models to the mean of waiting time for each port. The models are chosen with emphasis on getting four distinct approaches to predicting waiting time, namely linear regression, support vector machine, random forest, XGBoost and a feed forward neural network. Features and variables are pre-processed and hyperparameters are tuned for each model separately to achieve the best possible results. The data is split in to a training data set and a testing data set, with a 80/20 split, training and testing respectively. Moreover, 10-fold cross-validation is used when training the models on the training data. This allows for hyperparameter tuning without overfitting. The models are fitted to each port separately.

Model performance is evaluated using mean absolute error (MAE) and mean square error (MSE). If the prediction errors has outliers, MSE will penalize the model harder as each error is squared, as seen in Equation 3.10. By also using MAE (Equation 3.11), it is possible to evaluate the model with less emphasis on outliers.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2 \tag{3.10}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}| \tag{3.11}$$

3.5.4.1 Mean model

One of the simplest forms of prediction is used as a baseline model, the mean. The estimated waiting time ($\hat{y}$) for all vessels is the mean of n observed waiting times in the port. Shown in Equation 3.12.

$$\hat{y_i} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{3.12}$$

3.5.4.2 Linear regression

With linear regression, we fit a line to the data, defined in Equation 3.13, such that the sum of the residual sum of squares is minimized (Equation 3.14). In the case of multiple independent variables, a hyperplane is fitted. In the fitting process values for the intercept term ($\hat{\beta_0}$) and the slope coefficients ($\hat{\beta_1}...\hat{\beta_p}$) are estimated using the least squares method, shown in Equation 3.15 and 3.16 respectively.

$$\hat{y_i} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + ... + \hat{\beta}_p x_p \tag{3.13}$$

$$RSS = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.14}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.15}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{3.16}$$

When using linear regression, several assumptions about the data are made:

Linear and additive - The linear assumption states that the ratio in which the independent variable changes based on one units change in the independent variable remains constant. Also, in a multiple linear regression, with several independent variables, the association between the dependent variable and one independent variable should not be affected by the other independent variables. If the additive assumption is violated, the accuracy in the estimates of the regression coefficients are reduced. This situation is also known as collinearity.

Normality, independence and homoskedasticity - One key assumption when performing a t-test is that the data follows a normal distribution. This also holds true when calculating the significance levels of the regression coefficients. If the residuals are not normally distributed with a mean of 0, the inference about the true regression coefficients will be uncertain. Also, the variance of the residuals should remain constant, homoskedastic. If the residuals are not independent of each other, the calculated standard error tends to be underestimated. This leads to a lower p-value for the coefficients, which in turn implies more significant coefficients when this is not the case.

Three variants of linear regression are tested in this thesis, regular linear regression, ridge regression and least absolute shrinkage and selection operator (LASSO) regression, where the two latter are regularized models. This means that the models are penalized for model complexity. A penalty term is added to the loss function as shown in Equation 3.17 and 3.18 (Sohil et al., 2022). In practice this means that the coefficients of the least useful predictors will be set to 0 in the LASSO regression and close to 0 in the ridge regression. Regularizing a linear regression using these methods, will also help alleviate the effect of multicollinearity. The strength of the penalty is controlled with the hyperparameter λ.

$$\text{LASSO: } RSS + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3.17}$$

$$\text{Ridge: } RSS + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{3.18}$$

3.5.4.3 Support vector regression

Support vector regression (SVR) attempts to fit a straight line to the data, similarly to linear regression. Although the objective of both methods are similar, the methodology of the predictions are quite different. Support vector regression uses Equation 3.19 to make predictions.

$$\hat{y} = w^T x + b \tag{3.19}$$

Where:

w = Weight vector

T = Transpose operation

b = Bias term

In SVR, an error function is introduced, which is a margin around the fitted line. All errors within this margin are not considered errors, known as the ε-insensitive error function. This allows the model to generalize, and avoid overfitting. Fitting an SVR is an optimization problem where the objective function (Equation 3.20) is to minimize the model complexity, represented by the euclidean distance between each weight in the weight vector, and the sum of the error outside of the ε-margin moderated by a L2 regularization term C. The optimal C and ε are found using a grid search.

$$\min \frac{1}{2}||\mathbf{w}||^2 + C \sum_i L(y_i, \hat{y}_i) \tag{3.20}$$

Subject to:

$$y_i - \hat{y}_i \leq \varepsilon + \xi_i \tag{3.21}$$

$$\hat{y}_i - y_i \leq \varepsilon + \xi_i^* \tag{3.22}$$

$$\xi_i, \xi_i^* \geq 0 \tag{3.23}$$

The objective function is subject to three constraints. Where the first and second constraints ensures that the predicted value is within the error margin plus a slack term ξ. Whereas the third constraint ensures that the slack terms are positive.

Support vector regression models are also able to handle non-linearity using the kernel trick. The data is mapped to a higher dimensional space using a kernel function, allowing the fit of a straight line to the data (Smola and Schölkopf, 2004). Three kernels are tested on the data in this thesis, linear, polynomial and radial, along with a several combinations of ε and C.

3.5.4.4 Random forest

A decision tree uses all available predictors, finding the predictor and the split for each predictor that minimizes the sum of squared errors. Random forest only considers a subset of both observations and predictors for each tree. It uses a combination of bootstrapping and aggregation. This means that for each tree in the forest, a bootstrapped sample of the data is considered. A bootstrapped sample is a randomly selected sample from the data, with replacement, implying that the same observation can be seen several times in a subset. Furthermore, only a random subset of predictors is considered for each tree, typically $\sqrt{p}$, the square of available predictors. Each tree is a weak learner, but aggregating an entire forest of trees yields a strong flexible model that avoids overfitting. The prediction made is an average of all the individual trees predictions in the forest. In the case of classification, the prediction is the modal value of all the predictions (Sohil et al., 2022).

The random forest algorithm hyperparmeters tuned in this thesis are numbers of trees in the forest, maximum depth of the trees and minimum samples to perform a split.

3.5.4.5 Extreme Gradient Boosting

Extreme gradient boosting or XGBoost for short, is a tree based prediction model used for both classification and regression problems. While a traditional forest algorithm builds each tree independently and uses the average tree prediction as the model prediction, XGBoost fits each tree based on the error residuals of the previous tree. This method is known as boosting. It does so by starting out with an initial prediction value equal to the mean of the response variable in the root node. A split is made in one of the independent variables and a similarity score is calculated for each new node as seen in Equation 3.24. Then the gain is calculated for the split using Equation 3.25. Several splits from the root node are made and the split with the highest gain is retained. The retained leaves are split into further nodes until a limit is reached, such as maximum tree depth, minimum number of observation in the terminal node or the hyperparameter gamma (γ) exceeding the gain. Once a tree is complete, a prediction can be made by taking the output values from the tree as calculated in Equation 3.26, multiplied by the learning rate (lr) and adding this to the initial prediction. The prediction is refined by training additional trees, adding the output values from these to the prediction as presented in Equation 3.27.

$$Similarity\ score\ (SS) = \frac{(\sum Residual_i)^2}{n_{residuals} + \lambda} \tag{3.24}$$

Where:

$\lambda = regularization\ parameter$

$$Gain = Left\ SS + Right\ SS - Root\ SS \tag{3.25}$$

$$Output\ value\ (OV) = \frac{\sum Residual_i}{n_{residuals} + \lambda} \tag{3.26}$$

$$Prediction = Initial\ prediction + lr \times OV_1 + lr \times OV_2 + lr \times OV_n \tag{3.27}$$

The XGboost algorithm uses L1 and L2 regularization to avoid overfitting. L1, known as the alpha parameter in the model is analogous to Lasso regression penalty term, whereas L2, the Lambda parameter is analogous to Ridge regression penalty term, seen in Equation 3.26.

Although similar to random forest, the XGBoost algorithm has additional hyperparameters to tune. Since the algorithm uses gradient decent, a learning rate has to be set. As well as the minimum loss reduction parameter γ.

3.5.4.6 Feed forward neural network

The feed forward neural network is the simplest form of neural network with an input layer, an output layer and hidden layers in between. Each input node represent a feature in numeric form, this is passed forward to the hidden layer(s) where the input data is multiplied by weights w and a bias term β_0 is added. In turn, an activation function is applied, which in the case of a ReLU activation function (Equation 3.29), negative numbers are set to 0 and positive numbers are given a value according to the activation function. In the case more than one layer, the output from one layer is sent from one layer to the next, which in turn transforms the output from the previous layer. Finally, the output from the hidden layers is fed to the output layer which is the models prediction,

see Equation 3.28. In the case of waiting time prediction, the output layer consists of one node, namely the predicted waiting time. Using a fully connected neural network with several layers allows for capturing complex interaction between the input features. Fully connected meaning that every node in one layer is connected to every node in the next layer. The activation functions introduces non-linearity to the model.

$$\hat{y} = \beta_0 + \sum_{k=1}^{K} \beta_k g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j) \tag{3.28}$$

Where:

β_0 = Constant term term

β_k = Weight for output node k

w_{k0} = Constant term for node k

w_{kj} = Weight for input feature j in node k

$g(z)$ = Activation function, formula 3.29

$$g(z) = \begin{cases} 0, & \text{if } z < 0. \\ z, & \text{otherwise.} \end{cases} \tag{3.29}$$

A feed forward neural network is fitted to the data by a process called backpropagation. This involves adjusting the weights and biases such that a cost function is minimized. In this thesis, the cost function is MAE or MSE, as the response variable is continuous. In a neural network with several hidden layers and a sizeable number of neurons in each layer, the number of weights and biases gets large fast. Calculating all of these using brute force is not practical from a computational standpoint. Therefore, backpropagation is performed using stochastic gradient decent, which calculates the gradient of the loss function with respect to the weights and biases. This is used to adjust the weight and biases such that the cost function decreases.

The architecture of the neural network used in this thesis is based on a extensive grid search of a series of parameters shown in the list below. Several combinations from two to three layers using 8 to 16 neurons in each layers are tested. Upon initial testing,

layers with more than 16 neurons and more than three layers were found to consistently under-perform and therefore dropped from the final model tuning. In addition dropout layers to alleviate overfitting are used in all instances. The activation function used is ReLU.

- Number of hidden layers
- Number of neurons in the hidden layers
- Weight decay - a penalty parameter incentivizing the model to keep the weights for each neuron low.
- Dropout rate - randomly sets a set of neuron values to 0, reducing the dependency on individual neurons in the network.
- Learning rate - sets the step size for updating the models weight each epoch.

Each unique set of parameters are trained for 2 000 epochs (iterations). However an early stopping mechanism is used to stop training if no improvement in the loss function is observed within 100 epochs.

4 Results

4.1 Descriptive Statistics

Table 4.1 contains statistics generated from the event logs for each port. Visits show the number of *started* visits in berths and SPMs for the given port in a given year, as well as the total amount over the period. A visit started in a given year is not necessarily ended in the same year. Unique IMOs show the number of unique IMO numbers observed in the same periods. Lastly, the mean time in berth/SPM shows the average time spent in the berths/SPMs, as calculated by Equation 3.6.

Port Name & Year	**Visits**	**Unique IMOs**	**Mean Time in Berth/SPM (Hours)**
Al Basra (Total)	3575	900	47.4
2019	953	480	49.6
2020	807	441	48.5
2021	836	409	43.2
2022	979	481	47.9
Al Jubail (Total)	2652	1213	28.6
2019	631	449	27.7
2020	587	464	29.1
2021	719	504	28.2
2022	715	491	29.2
Fujairah (Total)	5196	1770	36.5
2019	1196	603	37.8
2020	1210	667	36.5
2021	1286	649	36.0
2022	1504	794	35.8
Ras Tanura (Total)	3957	1079	25.4
2019	907	518	24.9
2020	921	543	25.8
2021	1001	535	26.3
2022	1128	585	24.7

Table 4.1: Descriptive Statistics for Berths and SPMs.

From Table 4.1 one is able to gain a quick overview of the port specifics over the four year period. For instance, Al Jubail has by far the lowest amount of visits, but the second highest number of unique IMOs served in total. Fujairah, being the largest in terms of both visits and unique IMOs does not have the longest time in berth, which is held by Al Basra. Furthermore, the number of visits per year for all ports seem to have increased

overall during the examined period.

Figure 4.1 and 4.2, show the flow of vessel classes to the different berths and SPMs. It is apparent from these figures that the four ports handle their incoming traffic in somewhat different ways. One observes in Figure 4.1 that Al Basra has the most equal distribution, with all vessel classes calling at all berths. The other three ports seem to favour SPMs for ships of Suezmax size or larger, with Al Jubail being the most extreme. The figures also provide an overview of the composition of vessel classes calling to the different ports, where all ports except Fujairah, have a majority of VLCCs.

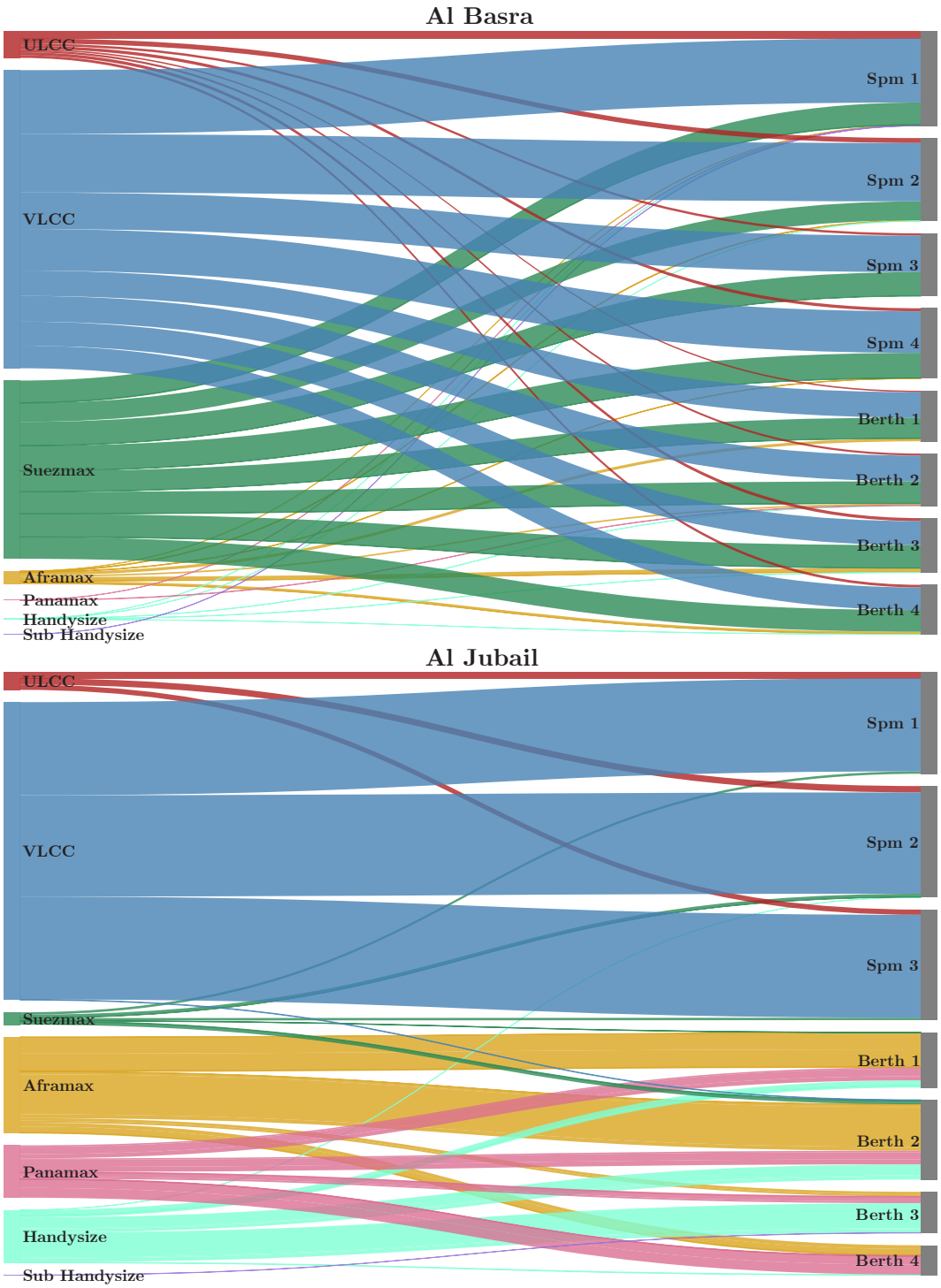


Figure 4.1: Sankey diagram of vessel flow for Al Basra and Al Jubail.

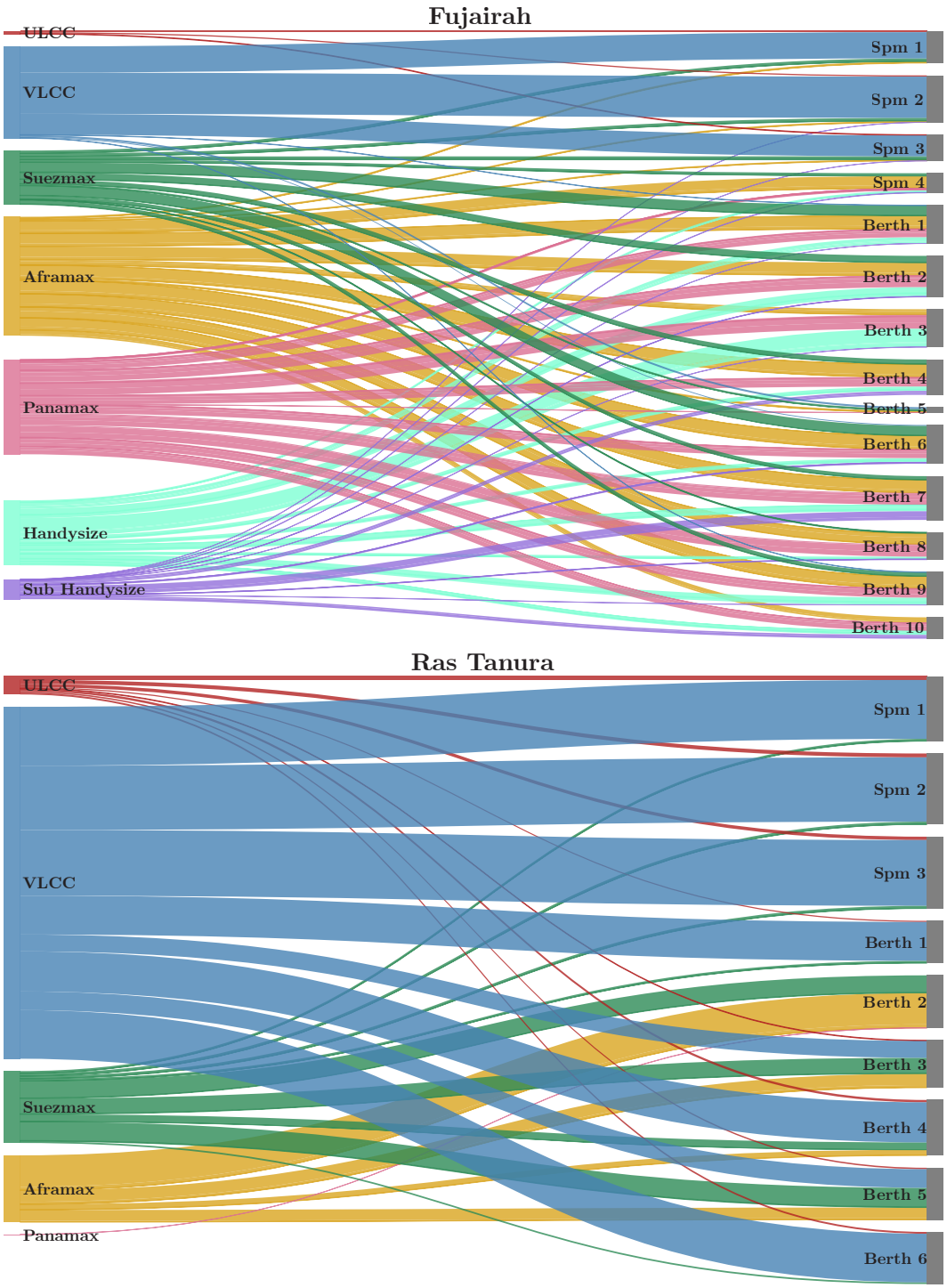


Figure 4.2: Sankey diagram of vessel flow for Fujairah and Ras Tanura.

Figure 4.3 is a representation of the different types (as classified by the STCS) of ships that made port calls to the four ports in our data. Although this plot is for a single year, it is representative for the entire four-year period in that there is little change year-on-year. These bar plots serve to underline the homogeneity in the types of ships observed in the data. Most of the observed ships are classified as *Crude Oil Tankers* with Fujairah and Al Jubail having the lowest amount of these ships with around 50-70%. This is expected, as the polygons are generated based on this type of ship. Over the entire four-year period for all four ports 77.83% of observations are *Crude Oil Tankers*.

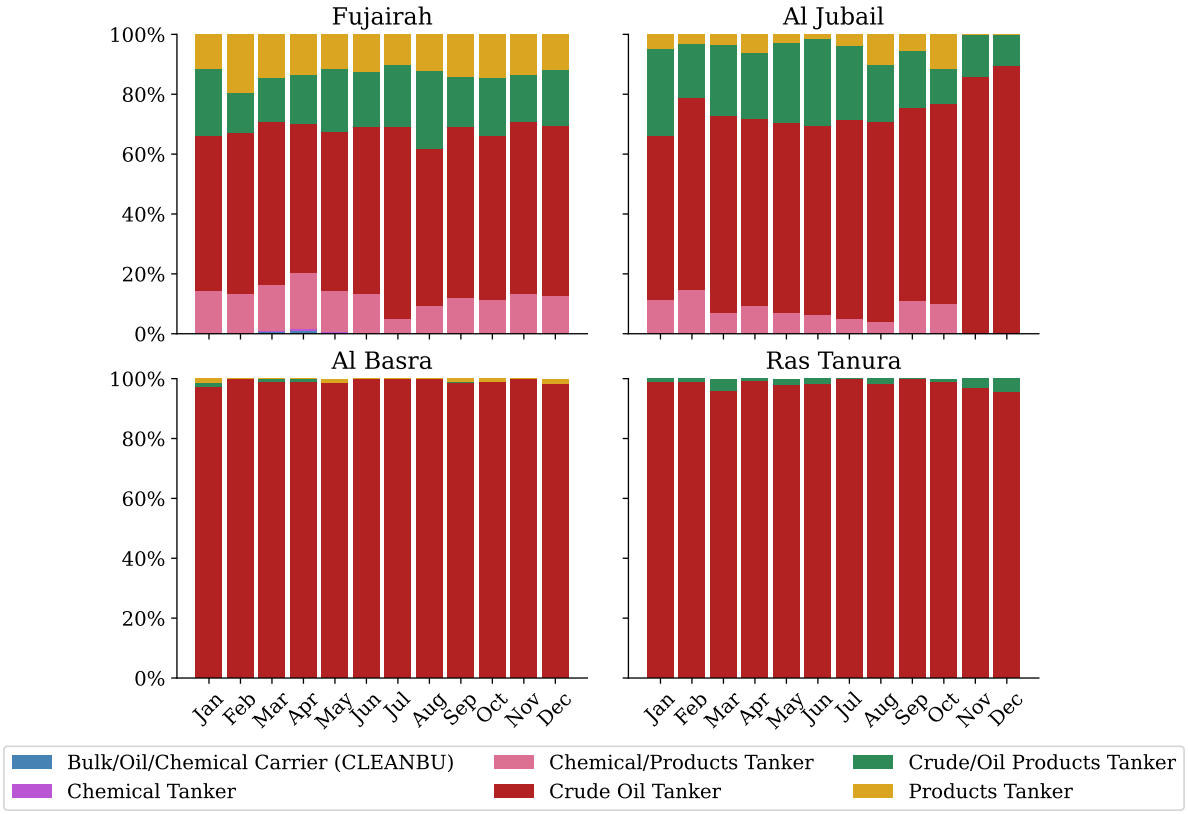


Figure 4.3: Percentage of ship types per month in 2022 for each port.

4.2 Validation Results

Illustrated in figure 4.4, the automated method of identifying port calls consistently underestimates the number of port calls. The main reason for this was discovered when plotting the track of the missing vessels. As the vessels approaches the berth, the AIS-track disappears before entering the polygon. Consequently, the trip is not accounted for in the automated data. Some vessels have well established routines for this, as a consequence

these vessels never show up in the thesis data.

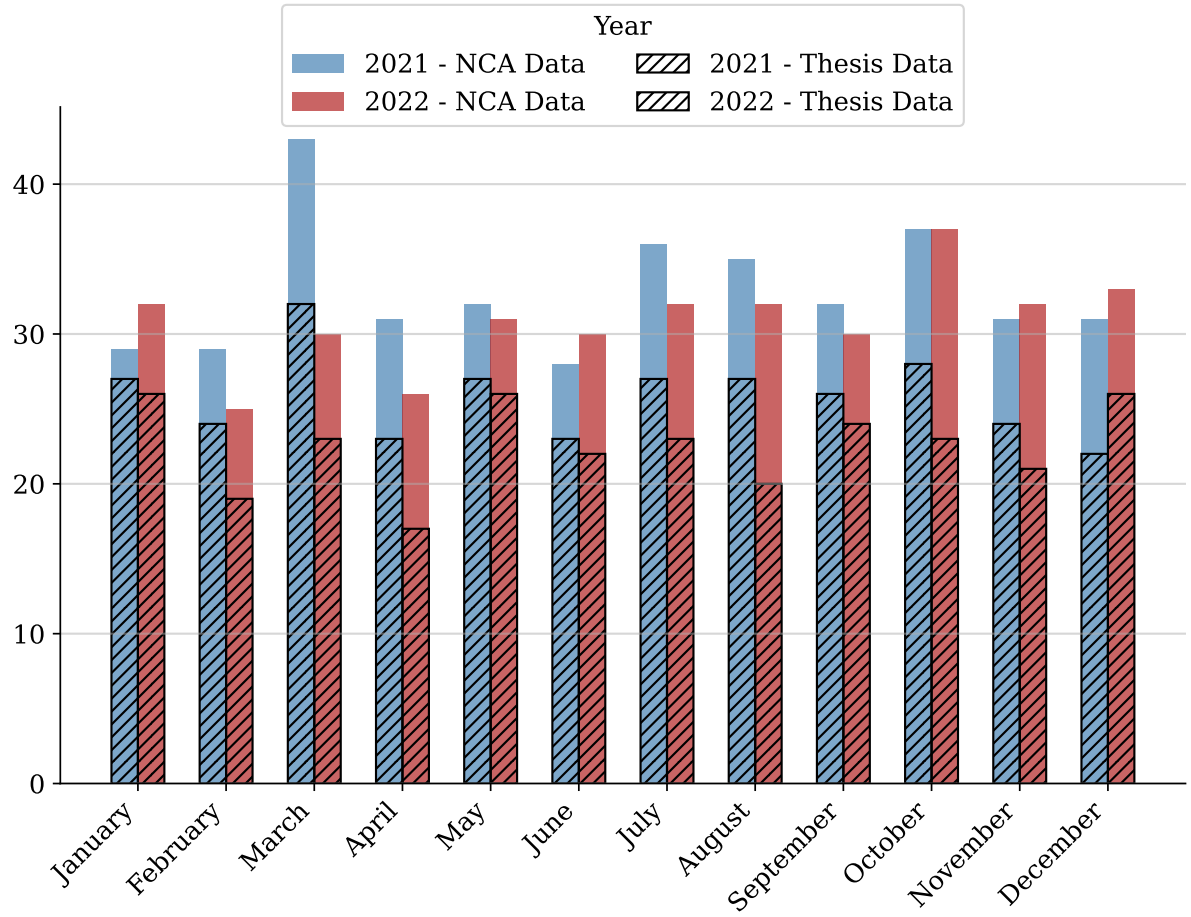


Figure 4.4: Count of trips as per NCA data and thesis data for the Norwegian oil terminals Mongstad and Sture.

The difference in berth time can be seen in Table 4.2. Worth noting is that the NCA data is based on self reported data from each vessel for each port call, with an arrival time and an *estimated* departure time. Whereas the thesis data is based on actual vessel movements. Note that vessels not present in the thesis data have been removed from the NCA data before further comparison. This is also the case for Figure 4.5.

	NCA Data	**Thesis Data**
Mean	1 day 03:00:07	1 day 05:05:41
Std	0 day 10:49:01	0 day 12:39:23
Min	0 day 02:30:00	0 day 02:28:08
25%	0 day 20:31:30	0 day 21:25:57
50%	1 day 00:18:00	1 day 01:17:11
75%	1 day 05:13:30	1 day 07:58:33
Max	3 day 21:24:00	4 day 23:27:56

Table 4.2: Summary of time at berth

Plotting the thesis data and the NCA data in a Gantt plot (Figure 4.5), illustrates the degree of overlap between these data sources.

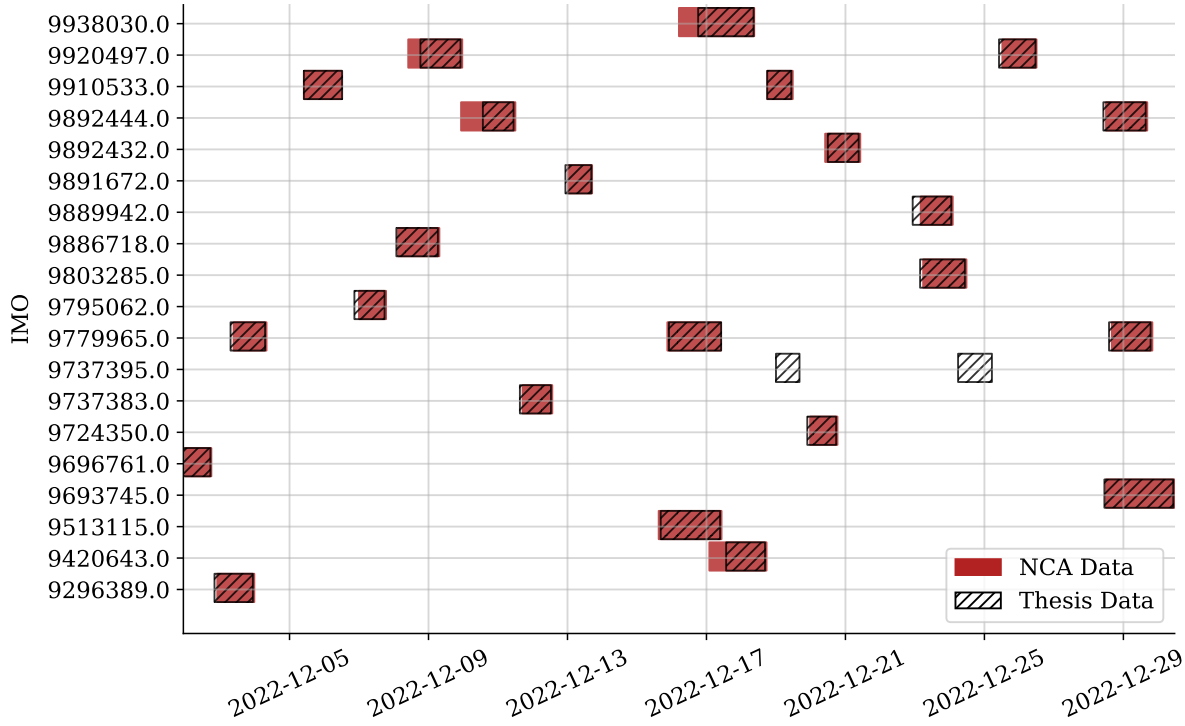


Figure 4.5: Crude oil tankers calling at Mongstad and Sture December 2022

4.3 Waiting Time Prediction

In this section, we will outline the distribution of waiting time across ports, look at performance of each model for each port, and examine a selection of the most important features for each model.

4.3.1 Waiting time distribution

Upon inspection of the dependent variable, waiting time, it is apparent that each port and each berth has its own distinct distribution. The Iraqi port of Al Basra consists of three SPMs and four berths in a sea island. The vessels calling at Al Basra are predominantly large vessels above Aframax size. All vessel classes are served by all berth positions, which means that once a berth position is available, any vessel can be served as seen in Figure 4.1. This is also reflected in the uniform distribution of waiting time for all the berth positions.

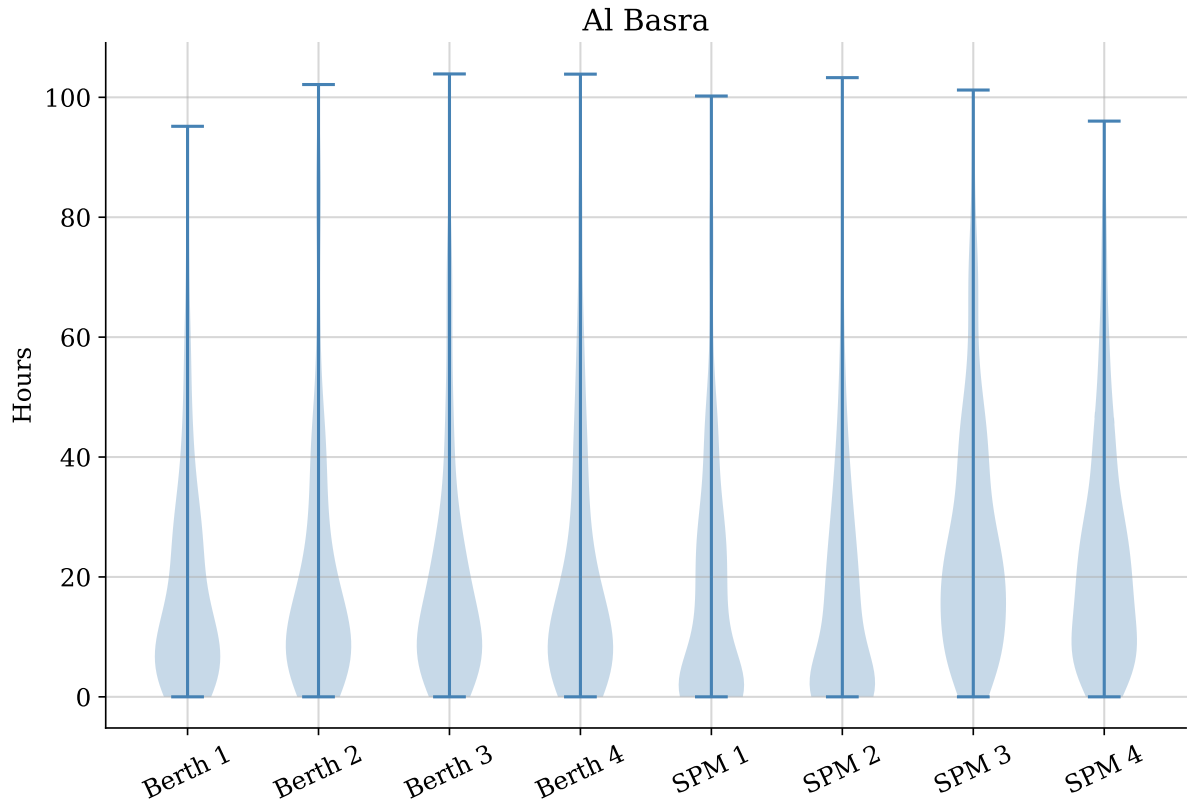


Figure 4.6: Waiting time Al Basra

Distinctly different from Al Basra, is Al Jubail in Saudi Arabia, where vessels of Suezmax size or larger are served by SPMs and all other vessels are served by land based berth positions. Practically no waiting time is observed in the SPMs whereas the variability in waiting time for the berths are significantly larger. Worth noting is that the number of port calls to the SPMs are larger than to the berths, see figure 4.1.

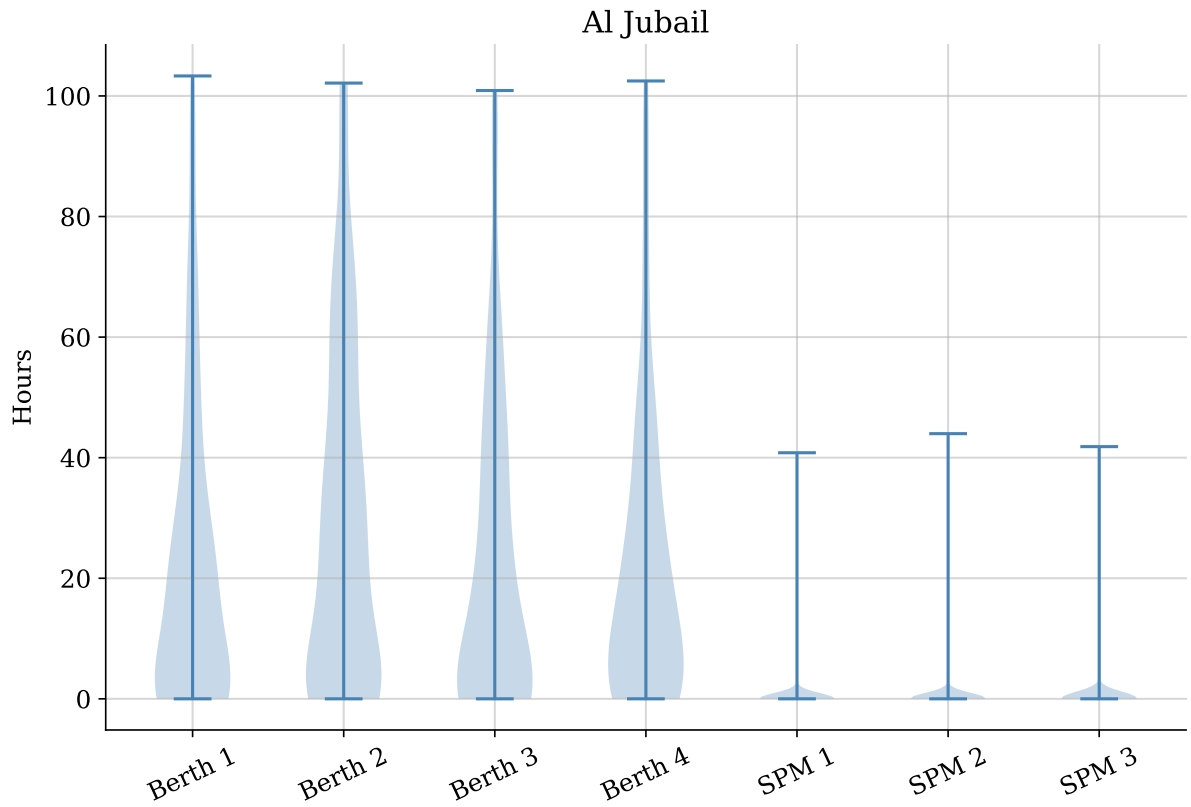


Figure 4.7: Waiting time Al Jubail

Fujairah is the busiest port with over 5 000 observations in our data. It is also the most diverse port in terms of the distribution of vessel classes. While some vessel classes are more frequent visitors than others, no class is significantly larger than the others. Furthermore, most berth positions serves a variety of vessel classes, except the SPMs that are dominated by VLCCs.

With a similar berth structure to Al Basra, Ras Tanura has a sea island with 6 berth positions along with three SPMs. Ras Tanurah predominantly caters for large vessels above aframax size.

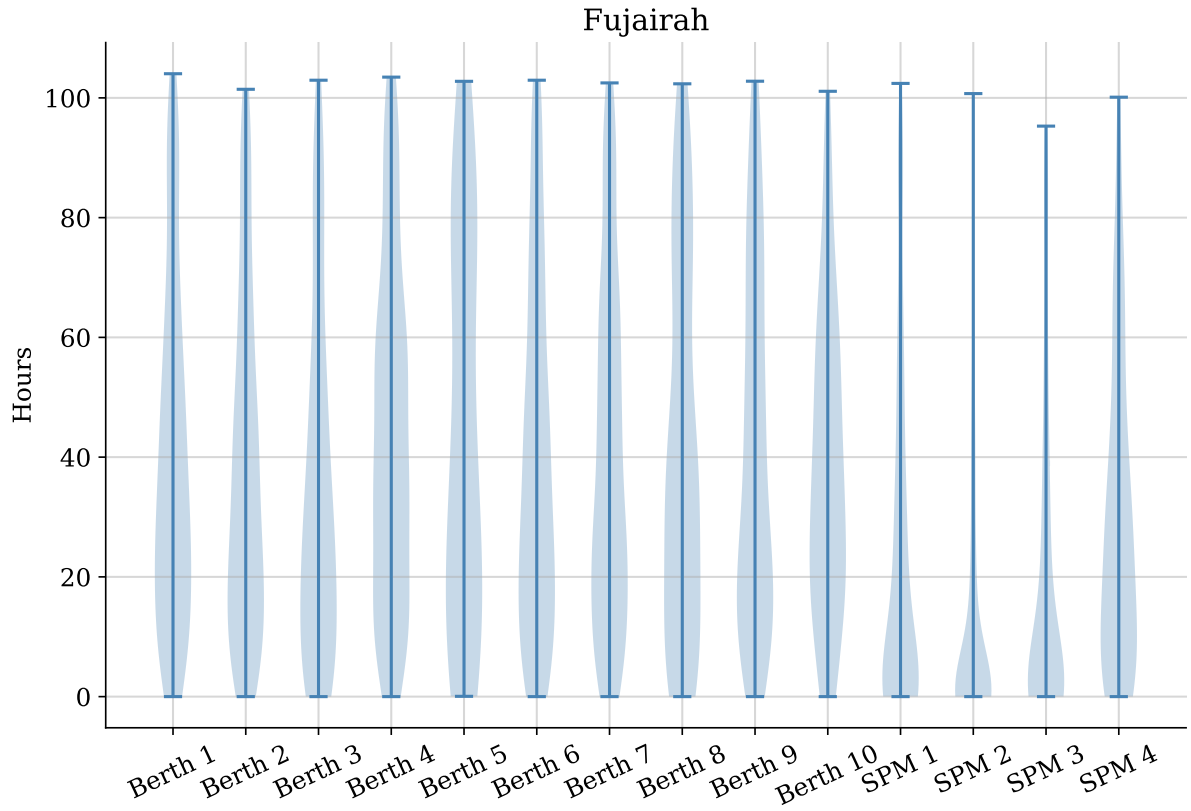


Figure 4.8: Waiting time Fujairah

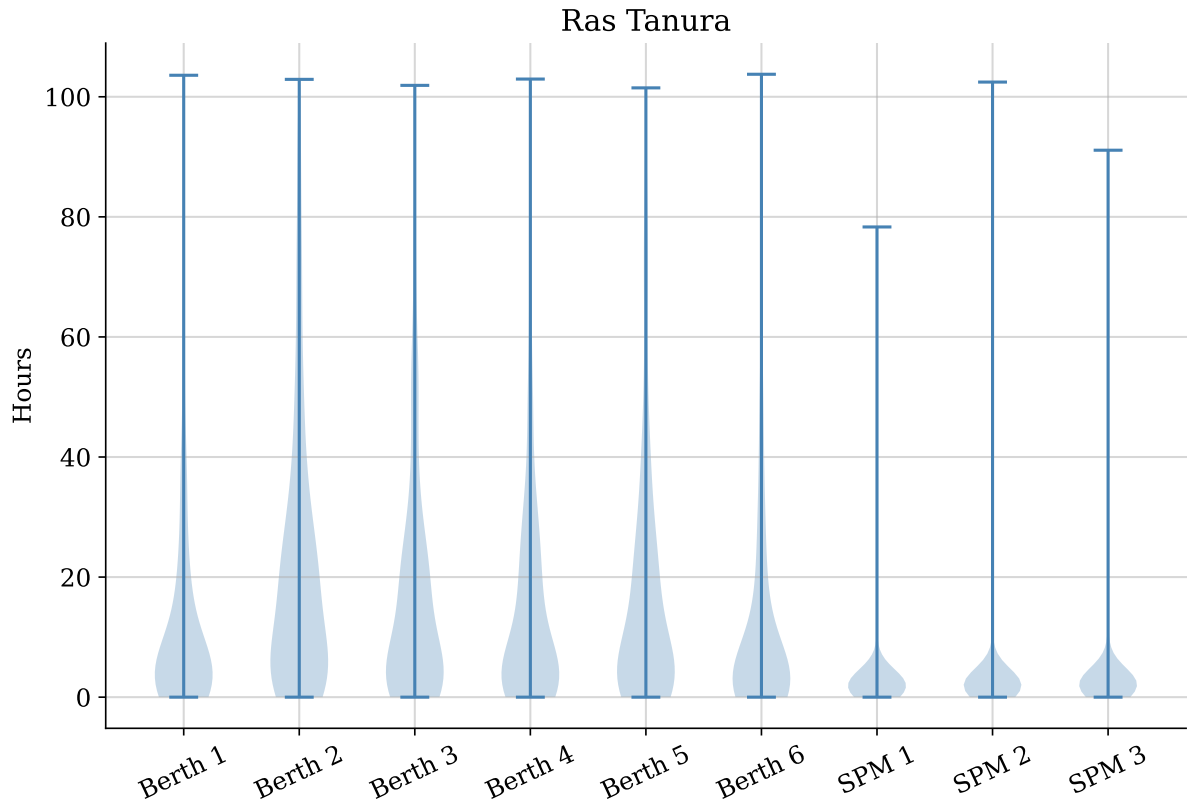


Figure 4.9: Waiting time Ras Tanura

4.3.2 Model performance

The tables below show test error in **minutes** for each model, for each port. In addition each models percentage increase in prediction performance as compared to simple mean is shown in the % Imp columns. Training time for each model in seconds is shown in the time column. Due to numerical instability, the linear regression model failed to converge on some ports. The date range from which the data was retrieved, along with n number of observations and p number of features can be seen in the bottom row of each table.

Model	**MSE Test**	**MAE Test**	**% Imp. MSE**	**% Imp. MAE**	**Time**
Simple Mean	116 325	210	0	0	0.01
Linear Regression	74 578	180	35	14	0.04
Lasso Regression	68 487	151	41	28	4.27
Ridge Regression	68 541	162	41	22	0.30
Support Vector Regression	104 946	129	9	38	40.43
Random Forest	62 290	120	46	43	406.41
XGBoost	82 253	117	29	44	232.06
Neural Network	73 623	127	36	39	404.15
Date range: 2019-01-01 - 2022-12-31				**p** = 85	**n** = 1 426

Table 4.3: Machine Learning Models Performance Al Jubail

For Al Jubail the best overall performing model is random forest, with consistent scores across both MAE and MSE. The neural network also performs consistent, but at a lower accuracy. During training the neural network showed signs of overfitting when performing considerably better on the training data than the testing data. The effect of L1 and L2 regularization can be seen when comparing the ridge and lasso regression against simple linear regression. Another consideration is the computationally efficiency, which varies vastly across the models.

There is a significant difference between Al Basra and Al Jubail in terms of model performance, both in absolute numbers and in improvement over simple mean. Less variability in performance across models is observed. The linear regression model failed to converge, which is likely due to numerical instability introduced by multicollinearity.

Model	**MSE Test**	**MAE Test**	**% Imp. MSE**	**% Imp. MAE**	**Time**
Simple Mean	941 876	783	0	0	0.01
Linear Regression	NA	NA	NA	NA	NA
Lasso Regression	764 214	660	19	16	1.06
Ridge Regression	767 811	662	18	15	0.35
Support Vector Regression	813 153	643	14	18	10.71
Random Forest	766 045	672	19	14	920.88
XGBoost	787 960	648	16	17	493.87
Neural Network	802 927	671	15	14	852.39
Date range: 2019-01-01 - 2022-12-31				**p** = 92	**n** = 3 575

Table 4.4: Machine Learning Models Performance Al Basra

Model	**MSE Test**	**MAE Test**	**% Imp. MSE**	**% Imp. MAE**	**Time**
Simple Mean	262 478	387	0	0	0.01
Linear Regression	NA	NA	NA	NA	NA
Lasso Regression	188 262	302	28	22	4.76
Ridge Regression	190 271	306	28	21	0.36
Support Vector Regression	237 969	268	9	31	12.51
Random Forest	167 808	254	36	34	926.96
XGBoost	161 928	241	38	38	470.54
Neural Network	207 455	284	21	27	859.26
Date range: 2019-01-01 - 2022-12-31				**p** = 86	**n** = 3 957

Table 4.5: Machine Learning Models Performance Ras Tanura

Model	**MSE Test**	**MAE Test**	**% Imp. MSE**	**% Imp. MAE**	**Time**
Simple Mean	13 184 484	2 806	0	0	0.02
Linear Regression	11 928 224	2 604	10	7	0.04
Lasso Regression	11 744 088	2 581	11	8	6.87
Ridge Regression	11 737 973	2 582	11	8	0.54
Support Vector Regression	12 944 756	2 412	2	14	53.84
Random Forest	11 143 786	2 501	15	11	2096.82
XGBoost	13 135 402	2 374	0	15	845.82
Neural Network	12 532 754	2 599	5	7	2010.26
Date range: 2019-01-01 - 2022-12-31				**p** = 111	**n** = 5196

Table 4.6: Machine Learning Models Performance Fujairah

Overall the best performing models are the regularized linear regression models, random forest and XGBoost. In terms of MAE, the support vector regression performs well across all ports, but is penalized by the MSE metric for some large errors.

A subset of the data containing only the three largest vessel classes, Suezmax, VLCC

and ULCC were also tested, which lead to a significant increase in model performance, especially for Fujairah. Table 4.6 shows the performance for the full data-set for Fujairah, whereas Table 4.7 shows the subset.

Model	**MSE Test**	**MAE Test**	**% Imp. MSE**	**% Imp. MAE**	**Time**
Simple Mean	7 130 745	2 046	0	0	0.01
Linear Regression	5 415 772	1 638	24	20	0.02
Lasso Regression	4 949 898	1 540	31	25	4.16
Ridge Regression	4 794 653	1 504	33	26	0.29
Support Vector Regression	5 511 955	1 458	23	29	2.78
Random Forest	4 423 777	1 415	38	31	625.35
XGBoost	5 874 329	1 439	18	30	315.27
Neural Network	5 830 166	1 628	18	20	602.57
Date range: 2019-01-01 - 2022-12-31				**p** = 111	**n** = 1 728

Table 4.7: Machine Learning Models Performance Fujairah large vessels

A prediction-truth plot reveals where the models are wrong, Figure 4.10 and 4.11 show the predictions from Table 4.3. Although the figure depicts predictions for Al Jubail, the general trend across ports is the same. Most models tend to overestimate predictions where the true value is 0, and underestimate predictions where the true value is in the higher end of the spectrum. For this particular port, the best performing model was random forest. Distinct for the regression models is that they output negative values for the waiting times in the lower end.

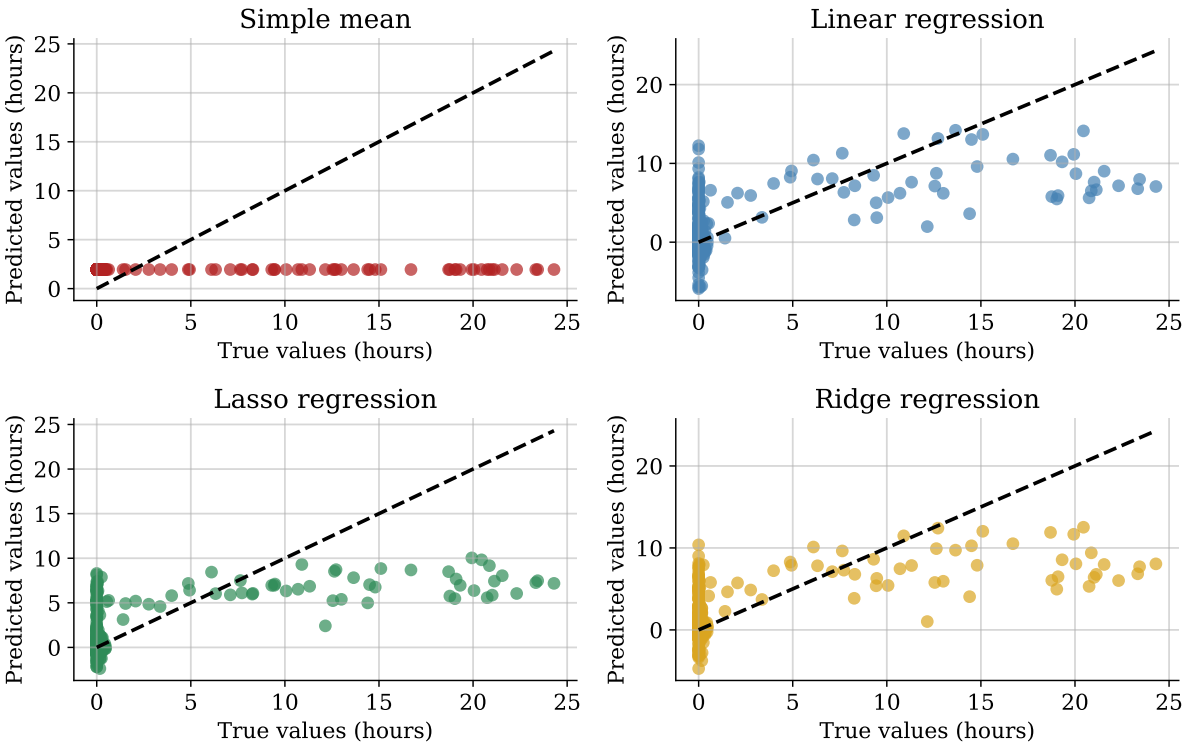


Figure 4.10: Predictions Al Jubail all vessels

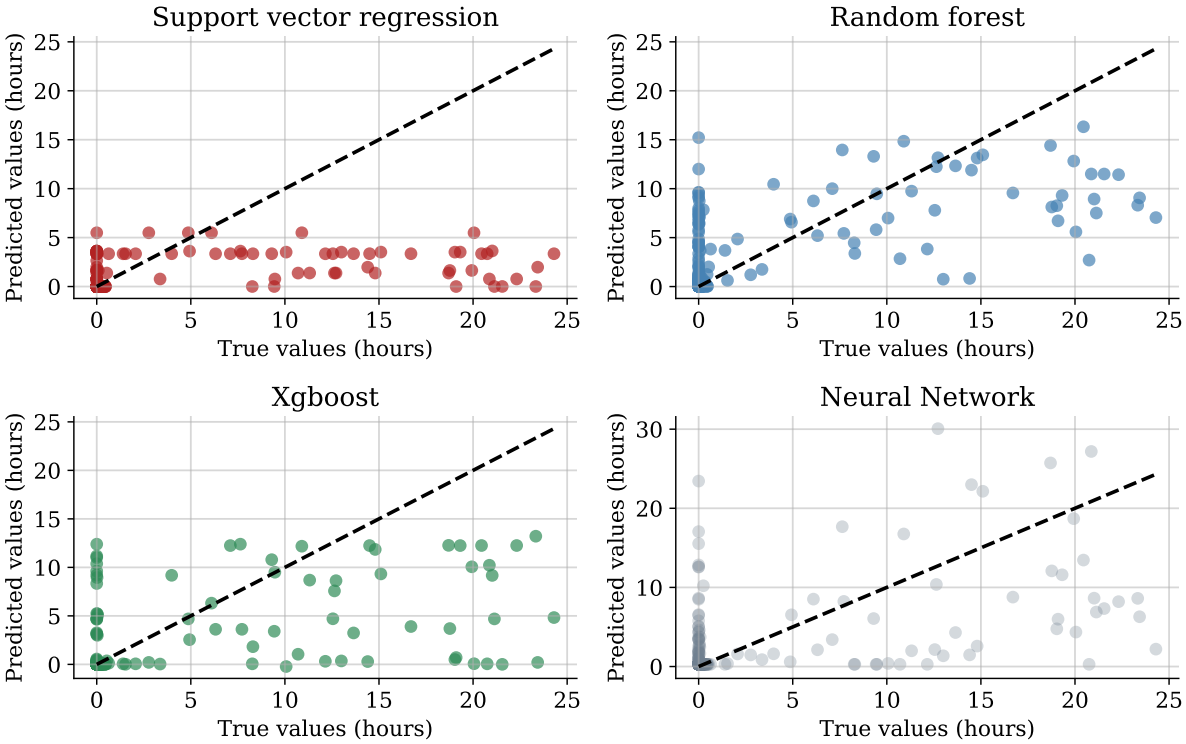


Figure 4.11: Predictions Al Jubail all vessels

4.3.3 Feature importance

Further insight is gained through looking at each features impact on the final prediction. The number of features ranges from 85 to 111, however, a small fraction of these are

dominating in terms of importance. Figure 4.12 shows the five most influential features for each port across all models. Since feature importance is determined in a different manner for each model, the weights, coefficients or feature importance values are normalized. Then the normalized values are summed across all models for each feature for each port.

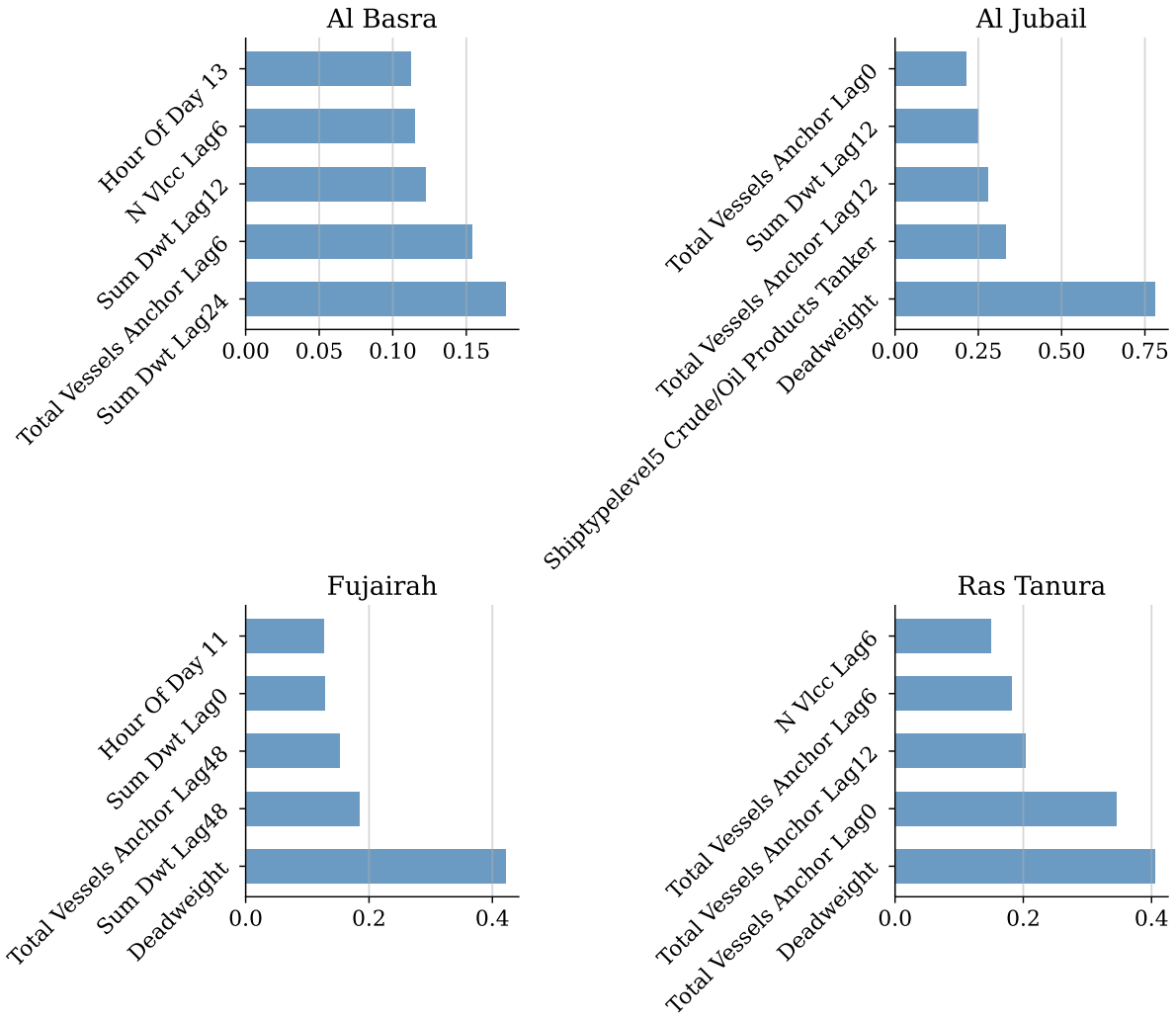


Figure 4.12: Feature importance for all vessels

Three categories of features are observed among the top five features, features describing the vessel calling, features describing the anchorage (queue) and features describing the time of day. Except for Al Basra, all ports have *Deadweight* as the most important feature, which is the carrying capacity of the vessel that the prediction is made for. Another feature that describes the vessel calling, is $Shiptypelevel5\ Crude/Oil\ ProductsTanker$, which is the STCS classification. The $Total\ Vessels\ Anchor\ Lag$ features describes the total number of vessels in the anchorage regardless of ship type with a given lag from when the calling vessel arrived at anchorage. Similarly *Sum dwt Lag* represents the total

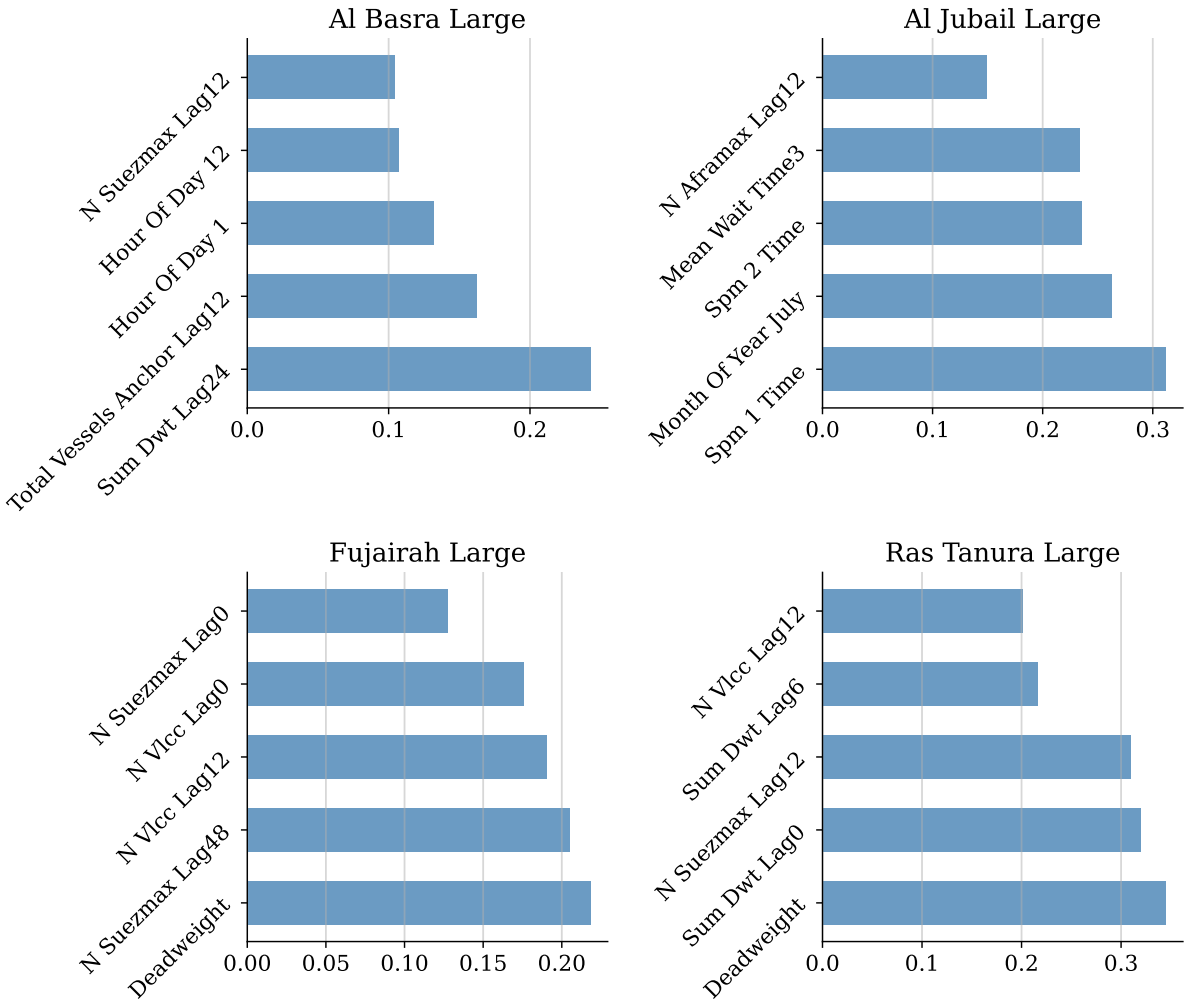


Figure 4.13: Feature importance for large vessels

tonnage in the anchorage at a given lag. For Al Basra and Fujairah, the hour of day of arrival appears to be significant variables.

When making predictions for vessels larger than Aframax, the important features changes, as seen in figure 4.13. The presence of large vessels in the anchorage represented by features such as $N\ Suezmax\ Lag$, $N\ Vlcc\ Lag$ and $N\ Aframax\ Lag$ have a larger impact on the final prediction. Two new features are also showing up in the top five list, $Spm\ 1\ Time$ and $Spm\ 2\ Time$ in Al Jubail. These features represents the time in berth elapsed for the vessel currently moored at SPM 1 and 2 in Al Jubail. Glancing at Figure 4.7, it is evident that there is little waiting time in SPM 1 and SPM 2 in Al Jubail. Once a vessel moores at one of these, it means that $Spm\ 1\ Time$ or $Spm\ 2\ Time$ is greater than 0, implying waiting time.

5 Discussion

5.1 Polygon Generation

Shortcomings of the methodology described in previous literature became evident when employed on AIS-data from tanker vessels. Most notably the lack of flexibility to cater for different mooring arrangements, and the insufficient ability to discriminate between ship to ship transfers and land based berths. We have shown that the use of port partitioning, shape factor and silhouette coefficient in tandem with DBSCAN mitigates these problems. Our method consistently performs well in accurately identifying port polygons for various ports and mooring arrangements. While it occasionally produces false clusters, this is more prevalent in complex ports such as Rotterdam and Singapore. In addition, one possible weakness with the method could be identification of SPMs in areas with strong prevailing winds, as this could cause the AIS clusters to form a half moon in stead of a circle. However, we did not observe this issue during our research.

For the most reliable results, the generation of polygons should be based on a time period close to the period the event log is generated. This ensures an overlap of polygons and actual activity for example if a berth is closed for maintenance. Another factor to consider is the size of the search grid. If the search grid is too small, there is a risk of missing out on potential anchorages or berths, and if the grid is too large, noise from other ports are picked up. Our assessment is that a reasonable middle ground has to be identified, based on the shipping segment of interest and geographical location.

5.2 Event Log Generation

The event log generated purely from raw AIS data (with the addition of ship types from the STCS) proves useful not only for predictions of waiting times, but also for developing insight onto port operations. The mean time in berth shown in Table 4.1, and the flow of vessel classes in Figure 4.1 and 4.2 points out differences in the ports. It seems that having designated areas for larger vessels, might contribute to more efficient port operations. Such information could lead to better informed decisions when choosing ports. Our findings also indicate that Fujairah is the most efficient of the examined ports, the mean time in

berth is lower than for the other ports when taking into account the number of visits. This is in line with findings by Merk and Dang (2012) who found scale to be an important factor in oil port/terminal efficiency. Such insights might prove useful for stakeholders in the shipping industry.

The results from the validation speaks to the robustness of the methodology used in this thesis. Given that the validation data partly relies on estimated times, the observed time differences in Table 4.2 stem from the accuracy of our systematic approach based on actual vessel observations. Although some vessels were not observed in the validation ports, we are still able to give an accurate estimation of the ports operations. At the very least it could provide better estimations of departure time for the terminals.

All vessels in our data are required to be fitted with class A AIS equipment, which have a nominal transmitting power of 12.5 watts (Shine Micro Inc., 2023). Ships that follow the International Safety Guide for Oil Tankers & Terminals (ISGOTT) , will reduce the power of their AIS transponders to one watt or less upon arrival to a terminal (ICS et al., 2006). This makes it difficult to pick up the AIS pings, and is likely the reason why there are missing ships in our data (when compared to manually collected data). The reliance on accurate AIS data is is therefore a limitation of our methodology.

Lack of data for validating the ports where waiting time is predicted is another limitation. When predicting, we are operating under the assumption that the data is predominantly accurate. However, without proper validation one cannot be certain of this. Obtaining data for validation is challenging due to the competitive nature of the industry, as safeguarding it could lead to competitive advantages. Regardless, we still believe the performed validation serves as a proof of concept for our method.

5.3 Waiting Time Prediction

The results from the predictions, shows that each port has its own dynamics and predictability of waiting times. We assess that some of the difference between ports stems from the composition of vessels calling at the port, and how these are served. This insight is gained trough an analysis of the most important features and descriptive statistics for each port. Al Basra stands out, since the most important feature for all the other ports, *Deadweight*, is not present. Our assessment of this is twofold, first, the vessels

visiting Al Basra are almost exclusively large vessels above Suezmax size. Second, berths are assigned regardless of vessel size or type. Therefore, four of the top five features for Al Basra reflect the number of vessels and amount of tonnage in the anchorage (queue). *Deadweight* is the most important feature for the other ports. This is likely due to the tonnage of the vessel calling to port deciding which berth it is assigned. In turn this affects expected waiting time, as this differs from berth to berth. Al Jubail is an example of this, where *Deadweight* is by far the most important feature, and large vessels are almost exclusively assigned an SPM, which has on average little waiting time. Another interesting observation is the presence of the predictor *Hour of Day*, which is in UTC. This was an important predictor for Al Basra and Fujairah. For Al Basra it was 16:00 local time and Fujairah 15:00 local time. This could indicate that vessels arriving at this time had longer waiting times. We believe that this is attributed to the working hours of the port staff. If a vessel arrives late, it must wait overnight, irrespective of the queue status.

Model performance across ports varies widely, with Fujairah being the most difficult port to improve over a simple mean prediction. The best performance was observed for Al Jubail where random forest yielded an improvement in mean square error of 46%. Leaving out the smaller vessels, predictions improved considerably for all ports. To ensure a consistent comparison, all models were tuned and fitted to each port separately. We assess that this variability is to be attributed to data quality, and the features ability to capture observed variability in waiting time, rather than model performance. In addition to this, some of the complex and flexible models tend to over-fit on the training data, despite regularization. This could indicate that a larger sample size would be beneficial.

In absolute terms the best performing model on the best performing port is random forest in Al Jubail, which yielded a mean average error of 120 minutes. The model´s prediction error ranges from close to perfect to being off by 15 hours. Whether this is an acceptable variability depends on the potential implications should the prediction prove wrong. Our overall assessment of the models across the ports, is that the precision yielded is not good enough for high stake decisions. However, they are a substantial improvement over using simple mean.

6 Conclusion & Further Research

6.1 Conclusion

We set out to answer if waiting times in crude oil ports can be predicted based on AIS-data. By answering this, new light can be shed on numerous facets of congestion in wet bulk terminals. From insight gained through analysis of model features contextualized by descriptive statistics, to operational decision support from the predictions itself. These insights are useful for both the industry and academia. Our main findings include that features extracted from AIS-data has predictive power on waiting time, that vessel composition and port dynamics correlates with waiting time, and that the expected waiting time across vessel classes and ports differ significantly.

We have contributed with an improved methodology in defining berth polygons from AIS-data, shed light on the reliability of data extracted with this method through validation, and increased the understanding of which features drive congestion in a wet bulk port. The practical implications depends on the user. For shipowners, the most profound application would be improved vessel economics, as knowledge about port congestion allows for speed adjustment. In addition, a precise estimate of vessel unavailability due to congestion will ease planning for maintenance as well as strategic fleet allocation. Academics are a step closer to a robust automated method for extraction of port data for further academic analysis. Other stakeholders in the maritime industry will be able to take advantage of a less opaque industry, as open data sources combined with automated data handling techniques reveals previously proprietary information.

6.2 Further Research

Based on the knowledge gained when working with this master thesis, we propose a set of topics for further research. Further refinement of the polygon generation methodology is needed to avoid false clusters. In addition, further research on polygon generation for different shipping segments and berth arrangements enables deeper insight into the segment specific hurdles, as with SPMs in the tanker segment. Our hypothesis is that each berth arrangement has distinct polygons and challenges, and would require methodological

adjustments.

The method used for identification of individual port calls warrants further investigation. We put forth a set of events defining a port call, and implemented this with code. Further research should look at fine-tuning the approach, taking into account the challenges encountered and highlighted in this thesis. Such as handling the splitting of trips, or gaining access to validation data for ports where predictions are taking place. The validation performed in this thesis made it clear that future research should also look at how to handle vessels with a reduced transmitter power.

All predictions in this thesis were based solely on AIS-data enriched with technical vessel data. A natural extension of this work would be to firstly, refine the set of extracted features through a deeper investigation of these, and secondly enhance the predictions using other sources of data. Examples of other data sources could be weather data, freight rates, commodity prices or world fleet data such as average vessel speed. Furthermore, a time series approach, as opposed to our cross sectional approach could prove beneficial in terms of predictive ability.

References

Abualhaol, I., Falcon, R., Abielmona, R., and Petriu, E. (2018). Mining Port Congestion Indicators from Big AIS Data. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2018-July.

Andersson, P. and Ivehammar, P. (2017). Green approaches at sea – The benefits of adjusting speed instead of anchoring. *Transportation Research Part D: Transport and Environment*, 51.

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software*, 22(4).

Benevento, E., Aloini, D., and Squicciarini, N. (2021). Towards a real-time prediction of waiting times in emergency departments: A comparative analysis of machine learning techniques. *International Journal of Forecasting*.

Data Mechanics (2022). What is data mechanics | data mechanics documentation. https://docs.datamechanics.co/docs/welcome. Last accessed Mar 17, 2023.

Dobrkovic, A., Iacob, M. E., and van Hillegersberg, J. (2018). Maritime pattern extraction and route reconstruction from incomplete AIS data. *International Journal of Data Science and Analytics*, 5(2-3).

Emmens, T., Amrit, C., Abdi, A., and Ghosh, M. (2021). The promises and perils of Automatic Identification System data. *Expert Systems with Applications*, 178.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *kdd*, 96(34):226–231.

Filom, S., Amiri, A. M., and Razavi, S. (2022). Applications of machine learning methods in port operations – A systematic literature review. *Transportation Research Part E: Logistics and Transportation Review*, 161:102722.

Fuentes, G. (2021). Generating bunkering statistics from AIS data: A machine learning approach. *Transportation Research Part E: Logistics and Transportation Review*, 155.

Google (2023a). Satellite Imagery of 25.2067407, 56.3707033. https://www.google.com/maps/@25.2067407,56.3707033,1024m/data=!3m1!1e3. Last accessed Apr 26, 2023.

Google (2023b). Satellite Imagery of 26.6681489, 50.1829361. https://www.google.com/maps/@26.6681489,50.1829361,2453m/data=!3m1!1e3. Last accessed Apr 26, 2023.

Google (2023c). Satellite Imagery of 26.9507227, 50.0470352. https://www.google.com/maps/@26.9507227,50.0470352,711m/data=!3m1!1e3. Last accessed Apr 26, 2023.

ICS, OCIMF, and IAPH (2006). *International Safety Guide for Oil Tankers & Terminals*. Witherby & Co Ltd, 5th edition.

International Mariime Organization (2019). AIS transponders. https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx. Last accessed Feb 03, 2023.

International Maritime Organization (2002). SOLAS - Chapter V Safety of navigation.

International Maritime Organization (2020). Just In Time Arrival Guide - Barriers and Potential Solutions. *GloMEEP Project Coordination Unit International Maritime Organization*.

International Telecommunication Union (2014). Recommendation ITU-R M.1371-5 (02/2014): Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band. https://www.itu.int/rec/R-REC-M.1371-5-201402-I/en.

Johnson, H. and Styhre, L. (2015). Increased energy efficiency in short sea shipping through decreased time in port. *Transportation Research Part A: Policy and Practice*, 71.

Kyritsis, A. I. and Deriaz, M. (2019). A machine learning approach to waiting time prediction in queueing scenarios. In *Proceedings - 2019 2nd International Conference on Artificial Intelligence for Industries, AI4I 2019*.

Kystverket (2023). SafeSeaNet Norway. https://www.kystverket.no/sjotransport-og-havn/safeseanet-norway/. Last accessed May 03, 2023.

McKinsey & Company (2023). Tanker | McKinsey Energy Insights. https://www.mckinseyenergyinsights.com/resources/refinery-reference-desk/tanker/. Last accessed Apr 30, 2023.

Merk, O. and Dang, T. T. (2012). Efficiency of world ports in container and bulk cargo (oil , coal , ores and grain). *Regional Development Working Papers*, (2012/09).

Millefiori, L. M., Zissis, D., Cazzanti, L., and Arcieri, G. (2016). A distributed approach to estimating sea port operational regions from lots of AIS data. In *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*.

Munim, Z. H., Dushenko, M., Jimenez, V. J., Shakil, M. H., and Imset, M. (2020). Big data and artificial intelligence in the maritime industry: a bibliometric review and future research directions. *Maritime Policy and Management*, pages 577–597.

National Geospatial-Intelligence Agency (2020). Nautical Publications. https://msi.nga.mil/Publications/WPI. Last accessed May 29, 2023.

Peng, W., Bai, X., Yang, D., Yuen, K. F., and Wu, J. (2022). A deep learning approach for port congestion estimation and prediction. *Maritime Policy and Management*.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(C):53–65.

Sanit-In, Y. and Saikaew, K. R. (2019). Prediction of waiting time in one-stop service. *International Journal of Machine Learning and Computing*, 9(3).

Shine Micro Inc. (2023). AIS (Automatic Identification System) Overview. https://www.shinemicro.com/ais-overview/. Last accessed Mar 02, 2023.

Silveira, P., Teixeira, A. P., and Soares, C. G. (2015). Assessment of ship collision estimation methods using AIS data. In *Maritime Technology and Engineering - Proceedings of MARTECH 2014: 2nd International Conference on Maritime Technology and Engineering*, volume 1.

Slack, B., Comtois, C., Wiegmans, B., and Witte, P. (2018). Ships time in port. *International Journal of Shipping and Transport Logistics*, 10(1):45–62.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression *. *Statistics and Computing*, 14:199–222.

Sohil, F., Sohali, M. U., and Shabbir, J. (2022). An introduction to statistical learning with applications in R. *Statistical Theory and Related Fields*, 6(1).

Steenari, J., Lwakatare, L. E., Nurminen, J., Talonen, J., and Manderbacka, T. (2022). Mining Port Operation Information from AIS Data. *Changing Tides: The New Role of Resilience and Sustainability in Logistics and Supply Chain Management–Innovative Approaches for the Shift to a New Era. Proceedings of the Hamburg International Conference of Logistics (HICL)*, 33:657–678.

Svanberg, M., Santén, V., Hörteborn, A., Holm, H., and Finnsgård, C. (2019). AIS in maritime research. *Marine Policy*, 106.

UNSD (2023). AIS Handbook - AIS Handbook - UN Statistics Wiki — unstats.un.org. https://unstats.un.org/wiki/display/AIS/AIS+Handbook. Last accessed May 28, 2023.

Wikipedia (2023). Quickhull. https://en.wikipedia.org/w/index.php?title=Quickhull&oldid=1145423573. Last accessed May 28, 2023.

Xiao, F., Ligteringen, H., Van Gulijk, C., and Ale, B. (2015). Comparison study on AIS data of ship traffic behavior. *Ocean Engineering*, 95.

Yang, D., Wu, L., Wang, S., Jia, H., and Li, K. X. (2019). How big data enriches maritime research–a critical review of Automatic Identification System (AIS) data applications. *Transport Reviews*, 39(6):755–773.

Zhang, L., Meng, Q., Xiao, Z., and Fu, X. (2018). A novel ship trajectory reconstruction approach using AIS data. *Ocean Engineering*, 159:165–174.