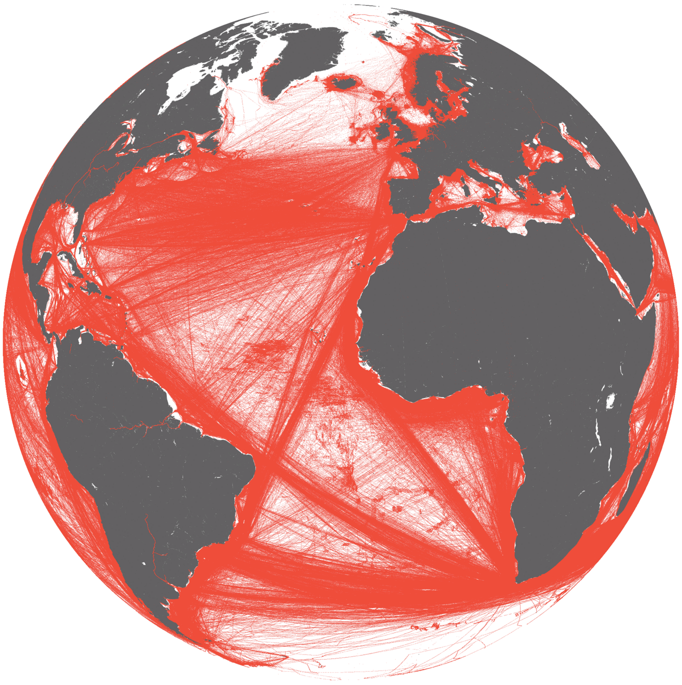
Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Morten Omholt-Jensen

Vessel destination forecasting based on historical AIS data

Master's thesis in Applied Computer Science
Supervisor: Christopher Frantz
June 2021



NTNU
Kunnskap for en bedre verden

MARITIME OPTIMA

Morten Omholt-Jensen

Vessel destination forecasting based on historical AIS data

Master's thesis in Applied Computer Science
Supervisor: Christopher Frantz
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Preface

This report serves as a Master's thesis in Applied Computer Science at Norwegian University of Technology and Science (NTNU) written in the spring semester of 2021. It explores topics surrounding vessel destination prediction for the purpose of maritime logistics. The author of the thesis has a prior Bachelor's degree in Programming (Applications) from NTNU, and the thesis was conducted in collaboration with the maritime technology startup company called Maritime Optima AS (MO) where the author is currently employed as a part-time developer.

Recommended prior knowledge

Since this thesis is written in the context of a Master's degree in Applied Computer Science, it is assumed that the reader has a background in computer science and is able to understand code examples, and is familiar with common programming terms, languages, and data structures. Furthermore, as the thesis focuses on Machine Learning (ML), it is recommended that the reader has an initial understanding of ML related, or statistical, concepts and methods.

Acknowledgements

This thesis would not have been possible without the help and support from my colleagues in Maritime Optima and their continuous support and feedback on this project. Our discussions, their knowledge, and their competence have all been valuable throughout the project.

A special thanks to the company's CTO Pål-Robert Engnæs for helping with data related discussions and problem formulation, and to the CEO Kristin Omholt-Jensen, for sharing her knowledge about the maritime industry and for opening up her network to me.

I would especially like to thank my supervisor Christopher Frantz for his excellent guidance and support throughout the academic process. His insight has helped me to understand the problem area in both an academic and commercial context.

I also want to thank other students, friends, and family for discussions, proofreading, and general feedback.

Lastly, I would like to give a big thanks to all the professors and fellow students at NTNU Gjøvik for 5 great years of experiences and learning.

Morten Omholt-Jensen

Abstract

The shipping industry is a vast and complex trading system that is capitally intensive, involves many companies and businesses, and is generally believed to be responsible for around 90% of all world trade (Tsaini 2011). Interested parties are all continuously searching for accurate information that can help them understand the future ebbs and flows of this volatile market that primarily consists of cargo demand and vessel supply. Thus, being able to effectively predict future movements and the availability of shipping vessels can be essential for many of the people involved in the industry.

Although the industry has traditionally relied on non-digital services, in recent years, there has been an increase in available software solutions that aims to assist shipping businesses in their decision-making processes. Many of these software products are based on the availability of Automated Identification System (AIS) data. AIS has become a globally adopted standard enforced by the International Maritime Organization (IMO) since 2006 for safety and navigation reasons. However, since AIS transmitters emit all commercial vessels' navigational data, it also has commercial value in that it provides a global overview of shipping vessels' movements over time. Recent studies into historical AIS data further elaborates that it is indeed applicable toward predicting future trajectories and movements of vessels and that Machine Learning (ML) techniques can be applied to this topic area.

This thesis investigates the area of vessel destination prediction and proposes a Machine Learning (ML) approach based on a combination of historical AIS data and technical vessel details such as vessel type, or segments. The proposed model applies to any vessel, is unrestricted by time or geographical limitations, and achieved an accuracy level of *72%* depending on vessel segments and subsegments. The thesis was written in collaboration with the maritime tech startup company Maritime Optima (MO) who provided the initial data foundation used to develop the proposed method.

Contents

Figures

Tables

Code Listings

Acronyms

AIS Automatic Identification Systems. viii, ix, xii, 1–8, 12–14, 16, 19, 21–23, 25, 26, 28, 36–44, 47, 54, 65–67, 79, 81–87, 89, 90, 92–95

APDE Average Prediction Distance Error. 27

COG Course Over Ground. 12, 26, 87

DBSCAN Density-based Spatial Clustering of Applications with Noise. ix, xii, 9, 23, 25, 38–41, 84, 90

DWT Deadweight Tonnage. 14

ETA Estimated Time of Arrival. 2, 12, 25, 27, 79, 93, 95

GIS Geographical Information System. 11

GT Gross Tonnage. 2, 12

IMO International Maritime Organization. 2, 12, 13, 36, 37, 54, 66

k-NN k-Nearest Neighbor. 23, 34

LNG Liquefied Natural Gas. 75, 77, 78, 88, 89

LPG Liquefied Petroleum Gas. 77, 78, 89

LSTM Long-Short Term Memory. 25

MAE Mean Absolute Error. 27

ML Machine Learning. iii, vii, x, 1, 4–6, 8–11, 15, 17, 18, 23, 25, 27, 35, 36, 47, 51, 52, 54–59, 63, 64, 66, 68, 84–86, 88, 91, 92, 94

MMSI Maritime Mobile Service Identity. 12, 13, 36, 54, 66

MO Maritime Optima AS. iii, ix, 3, 8, 13–16, 36–38, 54, 63, 66, 75, 77, 80, 83, 84, 93, 94, 104, 105

MSTD Most Similar Trajectory's Destination. vii, x, 35, 49–55, 57, 58, 65, 68, 72, 73, 75, 79, 85, 86, 90, 91, 94

NTNU Norwegian University of Technology and Science. iii, 22, 104

RF Random Forest. 9, 23, 25, 58, 59, 72, 86, 91

RMSE Root Mean Square Error. 27

RNN Recurrent Neural Network. 23

ROT Rate of Turn. 12

SMOTE Synthetic Minority Oversampling Technique. 18, 56, 57, 92, 95

SOG Speed Over Ground. 12, 26, 87

SSPD Symmetric Segment-Path Distance. 8, 9, 27, 50, 51, 53, 54, 68, 79, 84, 86, 91, 94

SVM Support Vector Machine. 23

XGBoost Extreme Gradient Boosting. xi, xii, 58, 59, 61–63, 65, 69, 70, 72, 73, 79, 85, 86, 95

Glossary

AIVDM/AIVDO The protocol used by AIS messages where AIVDM contains data received from other vessels, and AIVDO contains data from the owner vessel. A description of the protocol can be found at `https://gpsd.gitlab.io/gpsd/AIVDM.html`. xi, 12–14, 26, 35, 36, 38, 67

UN/LOCODE Five-letter geographic coding scheme maintained by the UN. The codes are assigned to, among others, ports where the first to letters represents a country code and the remaining three represents a location. 16, 37, 50, 55, 65, 66

vessel transition An event describing a change in a vessel’s naviagtional status emitted via AIS. A vessel is considered arrived at port when the status changes from “underway using engine” to “moored“ and departed from port when the status changes from “moored” to “underway using engine”. 14

voyage A journey a vessel undertakes at sea departing one port, and arriving at another port.. 1, 2, 6

Chapter 1

Introduction

1.1 Topics covered by project

The topics covered by this project mainly include applying computer science techniques to the problem of predicting shipping vessels' future destinations and voyage patterns to assist various actors in the shipping industry in their daily decision-making processes. More specifically, the thesis focuses on the aspect of applying Machine Learning (ML) techniques to vessel destination prediction using different sources of vessel information such as Automatic Identification Systems (AIS), voyage patterns, and individual vessel information such as vessel types, or segments. The goal of the thesis is to establish a high-quality, general prediction method not restricted by geographical extent or specific time intervals, and to discuss possible applications and value of the model in the current state of the art of the shipping industry.

1.2 Keywords

AIS data, vessel destination prediction, vessel supply, machine learning, maritime logistics

1.3 Problem description

Many, or most, companies in the shipping industry heavily rely on predicting the market in order to optimize their return of investment (ROI) and generally make smarter decisions resulting in beneficial investments. The market is generally defined by supply and demand where, in this case, demand consists of available cargoes to be shipped, and supply consists of vessels available to ship the cargoes. Figure 1.1 shows how different factors influence investment cycles within the shipping industry and the general market. Furthermore, the current methods for gathering data and conducting analysis are normally manual and paid services provided by specialists called brokers. The industry is still prone to using non-digital methods and external services to provide relevant information about vessel supply and traveling patterns.

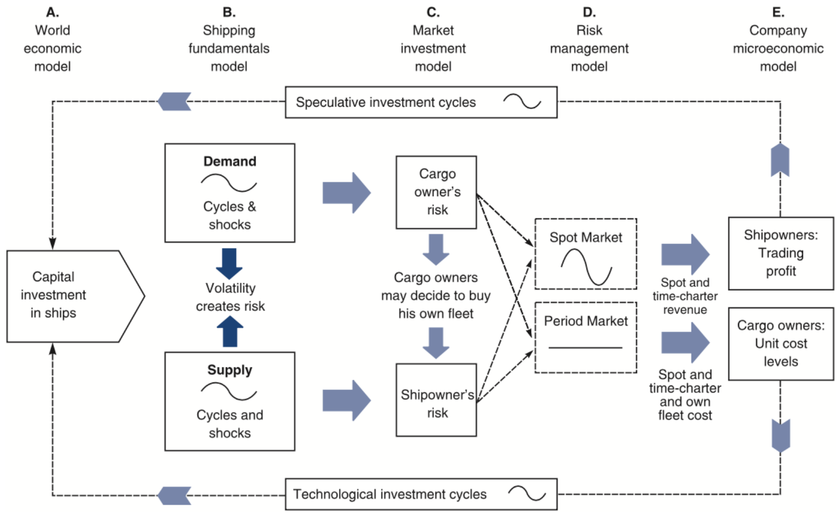

A.
World economic model
B.
Shipping fundamentals model
C.
Market investment model
D.
Risk management model
E.
Company microeconomic model
Speculative investment cycles
Demand
Cycles & shocks
Volatility creates risk
Supply
Cycles and shocks
Capital investment in ships
Cargo owner's risk
Cargo owners may decide to buy his own fleet
Shipowner's risk
Spot Market
Period Market
Spot and time-charter revenue
Spot and time-charter and own fleet cost
Shipowners: Trading profit
Cargo owners: Unit cost levels
Technological investment cycles

Figure 1.1: Vessel supply's role in the shipping market and investment cycles (Stopford 2008)

The data required to make good predictions are generally considered proprietary in the industry which is hesitant to share information. However, in recent years, vessel information has become more available through the AIS standard that provides information including vessel positions, navigational statuses, and manually inputted voyage information. In 2004, the International Maritime Organization (IMO) initiated the AIS protocol which all commercial vessels over 299 Gross Tonnage (GT) are required to use. This serves as a plentiful source of information applicable toward the analysis of vessel availability on a global scale.

Although the usage of AIS has been enforced and globally adopted, manually inputted data within the protocol lacks standardization. These attributes of the AIS protocol includes non-navigational voyage related information such as the vessel's intended destination and Estimated Time of Arrival (ETA). In contrast, the positional and navigational information within the protocol is automized, and therefore mostly accurate.

The manually inputted information is managed by the vessel's crew or captain and is therefore prone to human error in regards to either format or misinformation. Mestl et al. 2016 claims the accuracy of this information to be as low as 4% in certain areas. To use AIS data, existing prediction methods, therefore, only consider the geographical attributes which are automated including geographical coordinates, similar to that of GPS, speed, and heading. On the other hand, other aspects such as vessel type, dimensions, and draft, have extensively been overlooked in such methods which limits them in terms of accuracy when applied to a general range of vessels. Therefore, this thesis proposes an approach to vessel destination predictions that takes advantage of a broader range of vessel and voyage information to construct a reliable and generally applicable prediction method.

1.4 Justifications, motivation, and benefits

The shipping industry is a vast industry that affects the entire world. It is generally believed to be responsible for *90%* of all world trade (Tsaini 2011) but is also a massive contributor to global air pollution which negatively affects the environment (Wan et al. 2016). However, because of the ever-increasing global demand for products and services, it is presumable that the importance of the shipping industry will only increase in the future. This excludes reduction of shipping activities as a viable option, but it leaves room for innovation in terms of optimization since even small improvements on voyage routes and traveling patterns can have huge implications for both revenue and environmental impact. Furthermore, because of the vast volume of vessels and their cargo capacities, shipping investments generate a massive amount of revenue. For individual investors and companies, being able to rely on market predictions is key to making beneficial investments.

As an example, on 23 March 2021, one of the largest container vessels in the world, Ever Given, ran aground in the Suez Canal. This event was publicized worldwide because of the blockage's immense impact. Some estimates say the blockage cost on global trade lied between 6 and 10 billion USD[1], signaling the tangible impact of the shipping industry on the global economy as a whole.

Although there has been considerable research into vessel destination and trajectory predictions, the current literature appears to focus on smaller-scale predictions that emphasize topics such as collision avoidance and anomaly detection (Section 3.1). Furthermore, as mentioned in Section 1.3, existing works extensively overlook specific vessel details in favor of analyzing the geographical information provided by AIS. Of the research that offers more general predictions, such as forecasting the availability of vessels, efforts in this direction have been comparatively limited. The paper Lechtenberg et al. 2019 which was presented at the Hamburg International Conference of Logistics (HICL) in 2019 claims: *"Regarding the forecast of ship-supply so far — to the best of our knowledge — no research has investigated possibilities to predict the number of available ships in a certain region of interest."* which supports the observation made above.

To enable research in this direction, as part of this thesis, the collaborating company Maritime Optima AS (MO) provides high-quality historical AIS data in a highly available format and has already employed systems that can detect vessel arrivals and departures from a global set of shipping ports. This enables the thesis to focus more on analysis and applications rather than data collection and validation. Lastly, the thesis author has been employed at MO since the founding of the company and has been contributing to the development of their digital plat-

[1] https://www.bbc.com/news/business-56559073

form ever since. These factors combined are the main motivating factors behind this thesis.

1.5 Research questions

The main research question the thesis aims to answer is *"How can AIS data combined with specific vessel details be applied to predict future destinations of maritime vessels?"*. To successfully answer the main research question, more sub-questions are to be answered. Moreover, since the thesis aims to apply additional vessel information, mainly vessel segmentation, for the proposed prediction method, the final research question revolves around investigating the possible impact of this information on prediction methods. The full list of research questions are defined as follows:

1. How can AIS data combined with specific vessel details be applied to predict future destinations of maritime vessels?
   a. What prediction methods can be used to predict vessel destinations?
   b. What information can be used to predict vessel destinations?
   c. To what extent do methods proposed in existing work vary in scope of applicability?
   d. How can the validity of predictions made based on different prediction methods be established?
2. What is the impact of vessel segmentation by type, size, or capacity on prediction methods, or vessels' general predictability?
   a. What types of vessels are more predictable than others?
   b. Do larger vessels travel in more predictable patterns than smaller vessels?

1.6 Planned contributions

The main contribution of the thesis consists of proposing a generally applicable, global vessel destination prediction method that exceeds existing works' limitations to both geographical and time-related extent. The prediction method also takes advantage of a broader range of specific vessel details in an attempt to achieve higher general prediction accuracy for any type of vessel. The proposed solution includes a method of considering spatial trajectories as well as specific vessel details in a Machine Learning (ML) context. Moreover, the developed method provides a foundation that can be flexibly extended by adding more attributes about vessels or voyages to further explore their impact on predictions. To this end, the features used in the proposed solution are further investigated to determine their impact, or importance, and to determine relationships between features and predictability rates.

1.7 Remaining thesis structure

2. Background

This chapter aims to give the reader insight into the topic area in a technical sense as well as in the perspective of the shipping industry.

In this chapter, concepts, and terminology relevant to the thesis is explained including technological foundations such as Automatic Identification Systems (AIS), conceptual foundations such as AIS-based trajectories and trajectory similarity measurements, and techniques applied to predicting future destination ports of traveling vessels, namely, Machine Learning (ML).

3. Related work

In this section, related work and literature are presented and discussed in the form of a systematic literature review to establish the extent to which the current state of the art provides insight into the research questions listed in Section 1.5.

4. Methodology

In this chapter, the methodology of the proposed solution is explained in detail, as well as the development process and findings discovered when arriving at the proposed solution. This chapter is divided into sequential sections that each describe a step in the process used to compose the ML training dataset, the formulation of the analytical problem to be solved, and the Machine Learning (ML) related data preparation, training, and evaluation.

5. Results

In this chapter, the results from the proposed solution are described in detail. It describes the results from the different stages throughout the development process and presents the final results and metrics from the trained Machine Learning (ML) model. Furthermore, insights and interpretation of the results are gathered from shipping industry experts to qualitatively assess the validity of the proposed solution.

6. Discussion

In this chapter, a summary of the thesis is provided, followed by discussions regarding the proposed solution, the field of study, possible applications, and the approach's validity in terms of both academic and commercial values. Finally, limitations of the thesis and proposed future work are presented and discussed.

Chapter 2

Background

In this chapter, concepts, and terminology relevant to the thesis is explained, mainly, technological foundations such as Automatic Identification Systems (AIS), conceptual foundations such as AIS-based trajectories and trajectory similarity measurements, and techniques applied to predicting future destination ports of traveling vessels, namely, Machine Learning (ML).

2.1 Concepts

This section describes the broader concepts that are important to the thesis's solution and later discussions. The section's purpose is to give the reader a base understanding of conceptual foundations and challenges the thesis later refers to.

2.1.1 Vessel voyage definition

In order to effectively predict a vessel's future destination, or analyze voyage patterns in general, a vessel voyage must first be defined. This definition is in the context of constructing voyages from AIS data and is a crucial concept to define since it affects the outcome of any prediction method that considers historical voyages and ensures comparability with existing work within this area of study. The main factor to define is when a vessel arrives at a port, or more specifically, the conditions that must hold in order to consider a vessel as having arrived at a specific port.

There might be several different reasons for a vessel to visit a port, not all of which means that the port was the vessel's final stop in a voyage. For instance, larger vessels traveling long distances, often have to bunker (refuel) at bunker ports between the port they loaded cargo at and the port they eventually will unload the cargo at. In some cases, vessels anchor outside of such bunker ports awaiting to be refueled by bunker vessels, while in other cases they can reduce their speed and be refueled without ever stopping completely. Another common reason for vessels to physically stop moving is congestion in ports. Very often vessels of any size have to wait their turn before loading or unloading at busy ports. It is also common that vessels have to wait to pass through narrow canals. In these cases, they might anchor closer to a different port than the final arrival port while they wait for access. However, under such circumstances, vessels may not consider themselves “arrived” as they intend to discharge their cargo at a different port. In either case, whether vessels refueling at bunker ports, or stopping for other reasons, should be considered arrivals or not ultimately depends on the desired outcome of future predictions and context.

For the purpose of this thesis, an arrival is defined only when the vessel herself claims to be moored by reflecting this as a navigational status in the Automatic Identification Systems (AIS) data. As vessels usually do not use the moored signal when bunkering, or for short stops along a voyage, this entails that the proposed solution will be more prone to predicting the final destination of a vessel even though it might stop for other reasons along the voyage. This voyage definition is thought to be more beneficial for people working in the shipping industry who are interested in knowing what vessels are available in different regions for chartering. However, a disadvantage is that fewer voyages can be constructed from the available data as longer voyages could have been divided into multiple smaller voyages if considering bunkering, for instance, as port arrivals.

A literature study, later described in Section 3.1, shows that there are few studies that consider voyage prediction, however, the most common alternative method of defining trajectories of vessels is to use some form of clustering. The most promising of these studies defined port arrivals by detecting clusters of vessel positions transmitted close to ports. In contrast to using navigational statuses, this method defines voyages as trajectories between stopping ports, thus voyages stopping mid-voyage at smaller ports were considered separate voyages. The main advantage of this characterization is that the constructed voyages are more easily comparable as they do not include any additional port visits along its voyage trajectory. When compared to the aforementioned definition based on navigational statuses, there could be more voyages constructed using the cluster-based approach as it has a lower threshold for considering a port visit an arrival. Therefore, in the context of a prediction model, there would be more voyage samples available for learning when using the cluster-based definition.

As an example, consider a voyage starting in Brazil and ending in Shanghai, China. Depending on the speed and fuel consumption of the traveling vessel, this voyage is around 12 000 nautical miles long and would take between 30 and 40 days. Thus, it is probable that a traveling vessel would stop to refuel at a bunkering port such as the one in Singapore. In this example (shown in Figure 2.1), one could either consider one complete voyage from Brazil to China, or one could consider two voyages; one going from Brazil to Singapore, and another going from Singapore to China. Assuming the vessel uses the navigational status “moored” in Brazil and China, but not in Singapore, the approach used in this thesis would consider one complete voyage from Brazil to Singapore, since it reflects the intended voyage while a clustering-based method, in contrast, would consider the two shorter voyages.

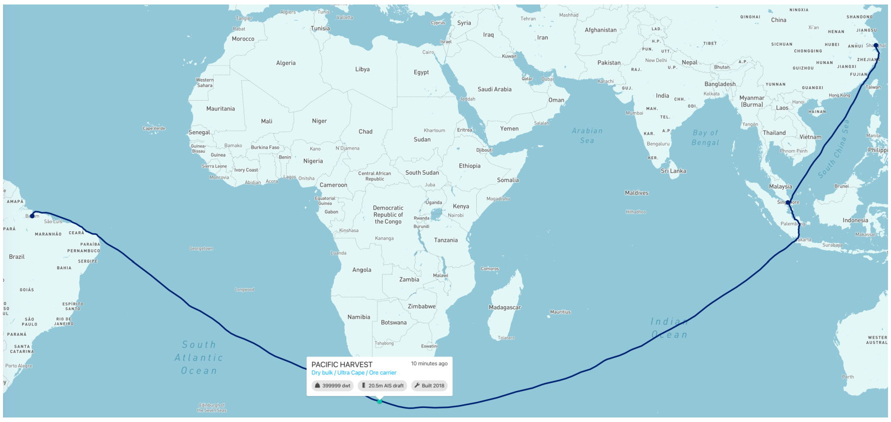


Figure 2.1: Example voyage, created using MO's route planner tool, for a traveling vessel (Pacific Harvest), traveling from Brazil to China while stopping at Singapore to refuel.

In this example, it is commercially more valuable for a prediction method to predict the vessel's destination to be in China rather than Singapore, since the fact that the vessel might stop to bunker at Singapore is somewhat obvious based on common sea lanes and voyages. This is the main reason for primarily focusing on the voyage definition using vessels' navigational statuses in this thesis.

2.1.2 Trajectory similarity

As will be further elaborated on in Chapter 3, the current literature related to vessel destination predictions almost exclusively relies on some form of trajectory similarity. Vessels' current trajectory seems to provide good insight into their intended destination since vessels are unlikely to follow unique trajectories during a voyage. Vessels are more likely to either follow established shipping lanes or the most optimal and fuel-efficient route. Trajectory similarity measurements can be used to find the most similar historical trajectory to the current traveling trajectory to predict where the vessel will travel to. Therefore, trajectory similarity is also included in this thesis' proposed approach to vessel destination prediction as a method of considering spatial information as well as vessel details.

There are three main categories of trajectory similarity measurements: spatial, temporal, and tempo-spatial. Regarding vessel trajectories derived from AIS, they are not likely to share similar time intervals values as vessels travel at different speeds and at different times. Therefore, for the purpose of this thesis, only spatial trajectory similarity measures are considered. This assumption is further corroborated by Zhang et al. 2020 that arrived at a similar conclusion in their work developing a ML -based approach to trajectory similarity measurements.

There are a number of spatial trajectory comparison methods that have been widely used for different purposes. The most relevant are the Hausdorff distance (Magdy et al. 2015), Fréchet distance (ibid.), and Symmetric Segment-

Path Distance (SSPD) (Besse et al. 2015). Out of these, the SSPD method is the most appropriate as it handles trajectories of different shapes and lengths well which is beneficial when comparing a trajectory from an ongoing vessel voyage to a set of complete historical ones. Figure 2.2 shows an example from ibid. where two trajectories are compared and their symmetric distances are calculated.

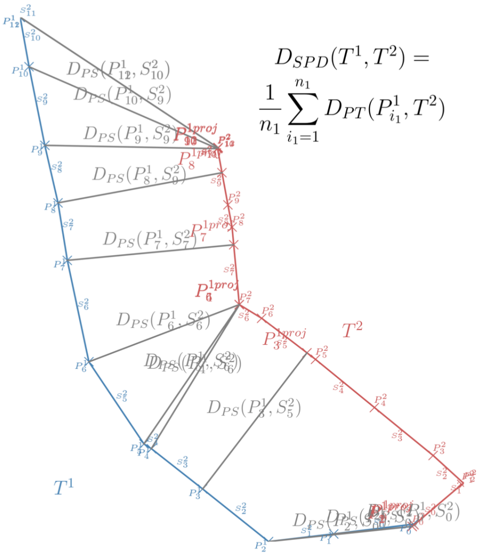


Figure 2.2: Segment Path Distance (SPD) in the SSPD process of comparing two different trajectories (Besse et al. 2015)

Moreover, the SSPD method is available as a convenient Python library that also supports different algorithmic similarity measurement methods. For these methods, a distance function can be specified and used to calculate the distance between points in the algorithm. This is important as the trajectories are specified as geographical coordinates, and as these are spherical in nature, the most appropriate distance function is the Haversine (Brummelen 2013) formula in contrast to the Euclidean formula commonly used for planar distances.

The methods mentioned thus far are all algorithmic approaches to measuring similarities between trajectories. However, there are also ML-based methods as well such as the approach proposed by Zhang et al. 2020 who also compare their results to the aforementioned methods.

They used a Random Forest (RF) model to measure trajectory similarity to find the most similar historical trajectory to any given traveling trajectory departing the same port. The most similar historical trajectory's destination is predicted to be the traveling vessel's destination. The study achieved a higher general accuracy level when compared similar approaches using algorithmic methods such as SSPD.

Moreover, some unsupervised clustering methods have also been applied to similar problems such as the Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm (Ester et al. 1996) which is capable of sequentially finding patterns in points and trajectories. This approach is more frequently used in trajectory predictions on a small geographical extent such as for collision detection and anomaly detection.

2.1.3 Machine learning (ML)

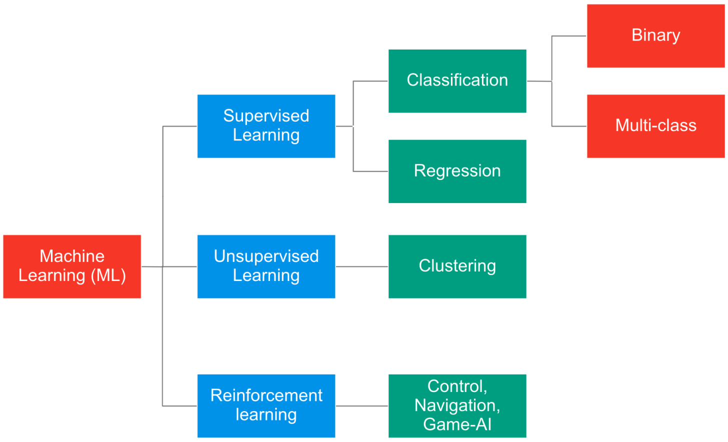


Figure 2.3: Machine Learning (ML) hierarchical terminology

Machine Learning (ML) is an umbrella term describing computer algorithms that automatically adapt and improve based on experience. Machine learning models are built based on a training dataset from which it derives patterns between underlying features. A trained model can be used to make predictions of a target value which can either be numerical or categorical.

There is a vast number of different ML algorithms applied to different problem areas. ML is mainly divided into three broad categories: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, in the training process, both input and the desired output are provided to the model. The model finds patterns and correlations between input and output data during the training process, and when the model is trained or fitted, it is capable of guessing output given only input.

In unsupervised learning, no output labels are provided to the model leaving the model to find patterns in the input set on its own. Clustering is an example of unsupervised learning as the model finds and labels patterns in input data without any external guidance. Reinforcement learning is a dynamic approach to ML where the model continuously learns while trying to achieve a goal. In this method, the model navigates a problem space, and the program rewards or punishes the model that tries to optimize for rewards. In regards to topics covered by this thesis, ML-based trajectory comparisons involve unsupervised learning, while predicting destination ports is supervised as the historical destinations are known.

Moreover, supervised learning can further be divided into regression and

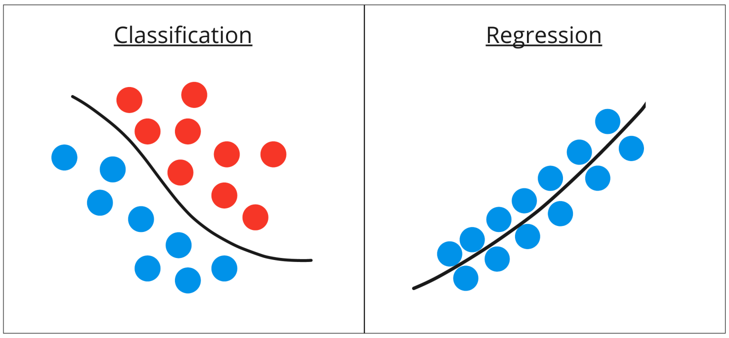


Figure 2.4: Example showing the difference between classification and regression tasks

classification problems. The main difference between the two is that classification aims at predicting a label, or a class, while regression predicts a quantity that is not necessarily present in the training data. For instance, a regression model can be used to predict the price of an item for sale, while classification can be used to label emails as "spam" or "not spam". Figure 2.4 shows the difference between classification and regression. The example of classifying emails as “spam” or “not spam” would be considered a binary classification problem as there are only two possible labels, however, classification can also involve predicting more than two outcomes which are commonly referred to as multi-class classification. In the context of this thesis, predicting a vessel’s destination port can be formulated as a multi-class classification problem as every possible destination port are different possible labels for a given voyage in progress. Figure 2.3 shows how ML is hierarchically divided into more specific terms relevant for the scope of this thesis.

2.2 Technologies and protocols

2.2.1 Database system

All the data that is used throughout this thesis for analysis is collected and stored in a *PostgreSQL* database. *PostgreSQL*, or *Postgres*, is an open-source object-relational database management system that supports the extended subset of SQL standards. One major advantage of using *Postgres* is the support for plugins such as *PostGIS* that provides tools for dealing with GIS and geometric data. In this thesis, *PostGIS* is frequently used to store and process geographical trajectory data for vessel voyages. Throughout this thesis, when referring to the proposed methodology and results, terms such as database, table, row, and column refer to the *PostgreSQL* database used and its tables with rows, and columns.

2.2.2 Programming languages and tools

The main programming languages used throughout this thesis are Golang and Python. Golang is primarily used in constructing the initial data foundation which requires dealing with databases, trajectory building, and validation. Golang is chosen for its performance benefits and ease of use. For data analysis and machine learning, Python is the main programming language of choice. Most code provided to the reader in this document is written with a focus on readability over efficiency.

2.2.3 Automatic Identification Systems (AIS) data

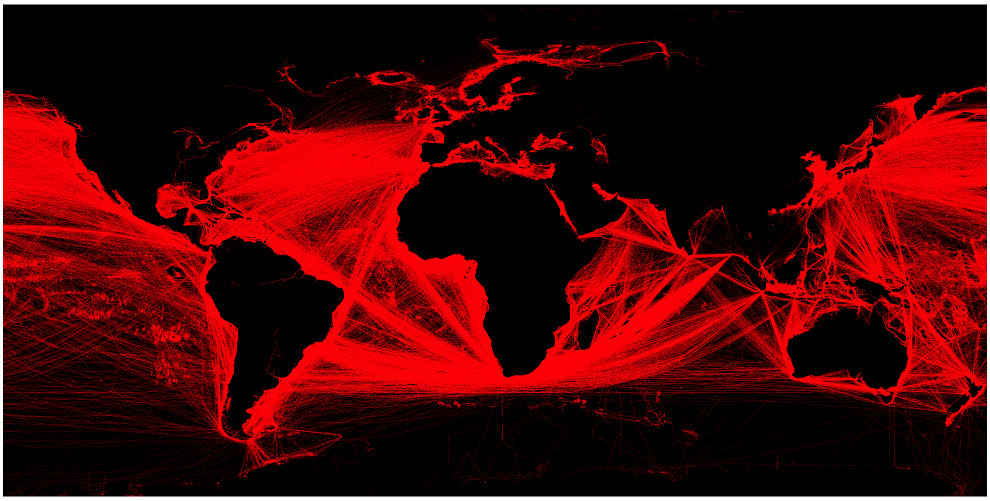


Figure 2.5: Vessel positions derived from 200 million AIS positional reports

As already mentioned in Section 1.1, Automatic Identification Systems (AIS) was initiated by International Maritime Organization (IMO) and since 2004 every commercial and passenger vessel exceeding 299 Gross Tonnage (GT) is required to carry an AIS transmitter. These transmitters broadcasts AIS messages following the AIVDM/AIVDO protocol. The AIVDM/AIVDO protocol contains two main types of reports: positional and static. The positional reports contains automatically collected information such as the transmitting vessel's Maritime Mobile Service Identity (MMSI) number, the current timestamp, and the vessel's current navigational data including the current geographical coordinates, Speed Over Ground (SOG), Course Over Ground (COG), true heading, Rate of Turn (ROT), and more. The static reports contain additional information about the vessel and its current voyages, some of which are input manually such as the vessel's IMO number, name, dimensions, draft, intended destination, and Estimated Time of Arrival (ETA).

As an example, Figure 2.5 shows a visualization of 200 million AIS randomly chosen positional reports from a collection of historical AIS positions for global collection of shipping vessels. In relation, the historical AIS dataset used in this thesis consists of more than one billion records ranging from December 2019 to March 2021.

Regarding vessel identification in the AIVDM/AIVDO protocol, there are mainly two values that are unique to a given vessel: the MMSI and IMO numbers. Either of these should be unique on their own for a given vessel, however, MMSI numbers can be recycled under certain conditions such as when a vessel is put out of commission while the IMO number is specific to a vessel's hull. Therefore, IMO is the preferred identifier, however, since the AIVDM/AIVDO protocol divides these identifiers into positional and static reports, both need to be considered in order to use both static and positional AIS information.

Since MMSI values can be recycled, a mapping between MMSI and IMO is required. Throughout this thesis, this mapping is provided by Maritime Optima AS (MO) and based on the latest combination of IMO and MMSI values found in the AIS data. This mapping is somewhat flawed as there could be different combinations between the same IMO and different MMSI values throughout the historical dataset. However, recycled MMSI is a rare occurrence in the *1.5* years of historical data provided, thus, the mapping is considered sufficient for the purpose of the thesis but could yield potential issues for a few number of vessels.

2.3 Initial data foundation

This section describes the form and meaning of the data that forms the foundation of the thesis' proposed solution. The data is provided by the collaborative company Maritime Optima AS (MO) to the author.

2.3.1 Vessel departure and arrival detection

MO collects live AIS messages provided by different sources, and in addition, they keep track of their navigational statuses as they are transmitted in the AIVDM/AIVDO protocol. These status attributes describe the current navigational state of the vessel for purposes of planning and security. Implicitly, these messages can indicate that a vessel has arrived or departed from a given port which can be used to detect voyages. When a vessel has concluded its journey and arrives at a port, the navigational status is changed to "*MOORED*", and when departing a port, the status is changed to "*UNDERWAY USING ENGINE*" or "*UNDERWAY SAILING*". There are also other navigational statuses that could be relevant for voyage information such as "*AT ANCHOR*" which could indicate that a vessel is bunkering (refueling) or is waiting for access to a berth that is congested.

Table 2.1 shows all the available statuses that vessels can emit in the AIVDM/AIVDO protocol. Currently, transitions from a status that indicates that a vessel is moving to the status "*MOORED*", and from "*MOORED*" to moving are collected and labeled as arrivals and departures from or to the closest port within a given radius. This has proven to be a sufficient method of identifying voyages

Status	Description
0	Under way using engine
1	At anchor
2	Not under command
3	Restricted manoeuverability
4	Constrained by her draught
5	Moored
6	Aground
7	Engaged in Fishing
8	Under way sailing
9–13	Reserved for future use
14	AIS-SART is active
15	Not defined (default)

Table 2.1: Available navigational statuses in the AIVDM/AIVDO protocol.

although it is dependent on the quality of the manually managed status value. Using this approach, voyages are defined by subsequent departure and arrival events, and positions between such events are collected as the voyage's trajectory. As this definition is based on transitioning navigational statuses, throughout this thesis, the concept is referred to as vessel transitions.

2.3.2 Additional vessel information and segmentation

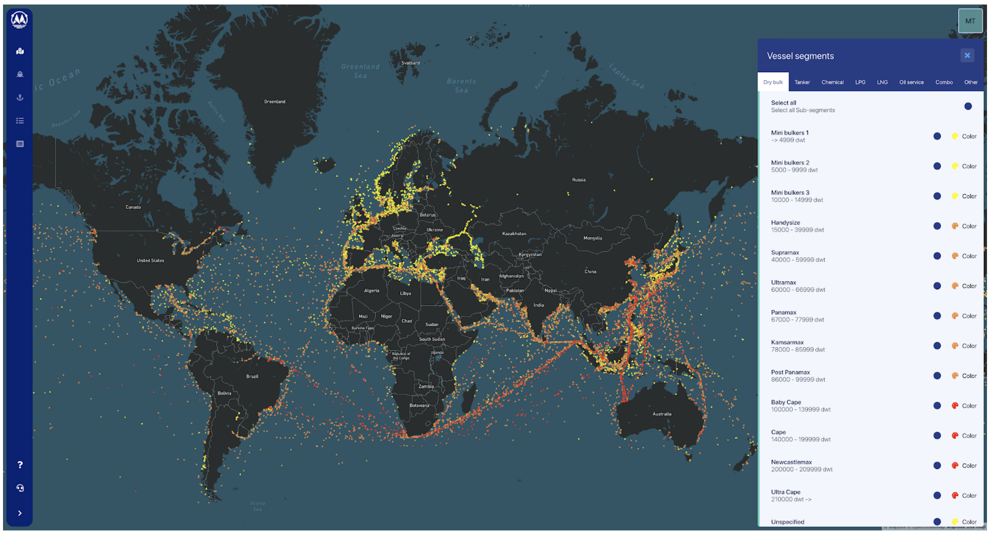
MO has implemented a system for categorizing vessels into different segments, subsegments, and further variations. These segmentations are based on various factors such as the dimensional data provided by AIS messages as well as technical vessel description provided by external sources and manual user input. One factor for defining vessels' segments can be found in the vessel type from the AIVDM/AIVDO protocol. However, the most important factor is input from external sources such as IHS Merkit[1] and DNV[2]. This provides a better segmentation than the values provided in the AIVDM/AIVDO protocol which only provides a much broader definition such as whether the vessel carries passengers, dry cargo, or is a tanker vessel.

For sub-segments, the most important inputs are cargo capacity and carry range, measured in DWT. This segmentation of vessels is highly relevant to voyage patterns as vessels of different types and sizes travel to different ports and countries for different shipping companies. This is further shown in Figure 2.6

[1] https://ihsmarkit.com/index.html

[2] https://www.dnv.com/

which shows, from an image of MO's web platform, how different sub-segments of the dry bulk cargo segment travels in different areas of the world. Since this categorization provides valuable insights into voyage patterns, vessel segmentation values are included in this thesis's proposed approach to vessel destination prediction.



Select all
Select all Sub-segments

Mini bulkers 1	-> 4999 dwt
Mini bulkers 2	5000 - 9999 dwt
Mini bulkers 3	10000 - 14999 dwt
Handysize	15000 - 39999 dwt
Supramax	40000 - 59999 dwt
Ultramax	60000 - 66999 dwt
Panamax	67000 - 77999 dwt
Kamsarmax	78000 - 85999 dwt
Post Panamax	86000 - 99999 dwt
Baby Cape	100000 - 139999 dwt
Cape	140000 - 199999 dwt
Newcastlemax	200000 - 209999 dwt
Ultra Cape	210000 dwt ->
Unspecified	

Figure 2.6: Maritime Optima AS (MO)'s segmentation of vessels where yellow vessels are smaller than reds

In addition to segment information, MO has done extensive work into gathering vessel information for their global collection of vessels via sources such as IHS Merkit and DNV. This information is publicized in their software solution where users can suggest changes to this public information which are validated by MO and applied if the information is valid. The extensive information collected for individual vessels creates a big potential for data analysis and developing ML models that are highly aware of specific vessels and how they travel. However, in this thesis, the main focus is on vessel segmentation when developing the proposed solution. This data is later referred to as vessel segments and includes both the vessels' segment and sub-segment.

2.3.3 Shipping ports

MO has an extensive port database containing more than 5600 ports. From sources such as UNECE, it is possible to find a vast number of ports, however, only a subset of the world's known ports are used by MO as these are considered relevant shipping ports. A port is deemed relevant if it offers loading, unloading, or bunkering services and has seemingly valid coordinates and identifiers. The process of determining what ports are relevant shipping ports is a continuous manual process in MO and it ensures that the available selection of ports is highly relevant for the industry.

Furthermore, all ports are identified by their UN/LOCODE. This is a five-letter unique identifier provided and managed by the United Nations (UN). In the five-letter code, the first two indicate the port's country of origin, while the three last indicate a more specific location within the origin country. As an example, the UN/LOCODE for the port of Oslo is `NOOSL` where “NO” stands for Norway, and “OSL” stands for Oslo. For comparison, a similar system is used for international airports. For this thesis, only the 5600 relevant ports that are considered relevant by MO's standards are used in the analysis.

2.4 Application challenges

Throughout the development process of the proposed solution, various implementation, or application challenges arose and were handled. This section aims to describe the background of these issues to help the reader understand the challenges and their respective solutions.

2.4.1 AIS data quality

One important issue to address is the quality of the underlying dataset. The AIS standard is globally adopted and enforced for commercial vessels, however, it lacks standardization for manually managed attributes. This issue affects the chosen voyage definition as it relies on the navigational status in the AIS data. For instance, if vessels neglect to change their signals, their defined voyage trajectories will not properly reflect a commercial voyage and might produce trajectories that are hard to compare with other historical trajectories. Based on manual inspection, vessels seem to be more consistent at changing their statuses from *underway using engine* to *moored* when arriving at a port, than the opposite when departing. This can lead to voyages beginning at some distance away from the departure port while ending more accurately at the arrival port.

AIS data transmitted is collected by either land-based base stations or orbiting satellites depending on the positions of the vessels. The geographical data transmitted is mostly reliable, however, as satellites have different orbits, there are gaps in their coverages. Vessels might travel up to several hours before a satellite collects their transmitted AIS data. There can also be issues with data transmitted in congested areas due to interference from other vessels. Some of these issues cannot be avoided yet affects the outcome of the work conducted in this thesis, however, some issues are identifiable in the historical data and can be managed for analytical purposes. For example, one issue with fluctuations in trajectories was identified and solved in this thesis as described in Section 4.3.2.

2.4.2 Categorical label encoding

Categorical values are values that are a subset of a finite number of possible values, while numerical values have infinite possible values. The thesis problem can be formulated as a multi-class classification problem since the predicted arrival port is a single value in a finite set of ports. ML models often perform better on, or expect, numerical values in their training datasets[3]. Therefore, it is common practice to use a form of encoding of the categorical values to transform them into numeric values. There are several different methods of achieving this, however, two common methods are “Label Encoding”, and “One-Hot Encoding”. In label encoding, each value in a category is transformed into a numerical value ranging from 0 to the number of unique values in the column. This is a simple and practical encoding method, however, since the categorical values are now numeric, implicitly, the ML might misunderstand the data to be ordered and derive meaning from the numerical relationships. In one-hot encoding, when a column of data is encoded, the column is split into multiple columns for each different category in the column. Then, the values are replaced by ones and zeros indicating what column contains that value. This solves the issue with implicit patterns in numerical values. However, with high cardinality datasets, ML models struggle with the sparsity and the large number of features that are generated, and can even perform worse than label encoding in some instances[4].

2.4.3 Imbalanced datasets and sampling methods

Imbalanced datasets are usually problematic in ML as models see more of certain samples than others making the model biased toward the more frequent outcomes. This can also lead to misleading accuracy values as, during the evaluation process, some values occur more often than others. For example, consider a binary classification problem where the arrival port is either port A or port B. If the dataset contains 90 samples where the arrival port was port A and 10 where it was port B, a simple function could be implemented that always predicts the arrival port to be port A without considering the input values and the “model” would have an apparent accuracy of *90%*. However, this accuracy would be misleading as the model will never predict port B as the arrival port. The same phenomena can occur in ML models that are trained on imbalanced datasets. Some ML models deal with the problem of imbalance better than others, especially decision tree ensemble methods such as the Random Forest (RF) model, however, these models might still struggle with highly imbalanced datasets.

There are several methods of dealing with imbalanced datasets including providing models with predefined weights, however, not all models provide this option. For a more general approach, it is possible to manipulate the dataset before training in an attempt to balance the classes. This can be achieved through sampling the dataset in a manner that produces a more balanced frequency of the outcome values. The two methods in question are minority oversampling and majority undersampling. Minority oversampling is a procedure where the minority classes, or target values, are duplicated while majority undersampling consists of

[3] `https://www.mygreatlearning.com/blog/label-encoding-in-python/`

[4] `https://towardsdatascience.com/d64b282b5769`

removing majority classes until the dataset contains an equal number of classes. On a general basis, these methods have their accompanying caveats. For instance, when oversampling the minority classes, the model has a greater chance of overfitting as it sees a high number of the same samples. In contrast, when undersampling majority classes important relationships in the datasets might be removed along with the samples of the majority classes.

There have been some attempts at minority oversampling without simply duplicating information. A popular approach called Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al. 2002) achieves this by synthetically generating new samples based on closely related samples in the dataset. For majority undersampling, Edit Nearest Neighbor (ENN) evaluates *k* nearest neighbors to find misclassified samples and removes them. This enables majority undersampling with less risk of removing important relationships in the dataset. Santos et al. 2018 describes possible implications that oversampling may have on imbalanced dataset evaluation. They found that datasets oversampled using SMOTE can lead to some misleading evaluation results such as that of an over-optimistic evaluation process. When any oversampling technique has been applied to a dataset, there is a risk of the evaluation sets containing many similar or duplicate values as the training set, thus, the model is evaluated using samples it has already seen during the training process. In general, they found that using a combination of over and undersampling using SMOTE and Tomek Links (Tomek 1976) respectively produced the most reliable results.

2.4.4 Model evaluation

After a Machine Learning (ML) model has been trained it must be validated properly in order to determine its real performance. The simplest evaluation process consists of dividing the full dataset into a training and a test dataset by a certain sample ratio, commonly *80%* train and *20%* test data. This is important as to not evaluate the model using samples it has already observed in the training process and to ensure that the model performs as expected on previously unseen samples. When the trained model performs well on the training data, but not on previously unseen samples, the model is overfitted. In such cases, the model essentially remembers the entire training dataset but has not learned it, so it cannot be applied to previously unseen samples. Moreover, to ensure that the selected portion of test data was representative of the dataset, *k-Fold cross-validation* can be employed (Ghojogh and Crowley 2019). In this process, the full dataset is divided into *k* parts of equal size, and the model is trained *k* times, using one different part as test data and the remaining as training data for each training process. The average accuracy value can then be extracted from each of the training rounds to determine a more balanced accuracy value. Furthermore, if the level of accuracy differs significantly per training round, or the standard deviation of accuracy is large, it indicates that the model can be overfitting in the training process.

Chapter 3

Related work

The topic of Automatic Identification Systems (AIS) -based predictions has already been explored quite extensively, especially in recent years as AIS systems have become an enforced standard for commercial vessels in the industry. However, the AIS standard has mainly been applied for the purpose of maritime safety and navigation, and the existing academic work on this topic reflects this. Most of the related work consists of vessel trajectory predictions for the purpose of foreseeing possible future collisions or for detecting anomalies from established shipping lanes. These types of predictions are applicable for predicting a vessel's future position in a short time interval, in a limited geographical area, but with high positional accuracy. In order to establish the current state of the art of the topic area and establish to what extent the literature answers the proposed research questions, a literature review was conducted which is described in the following section.

3.1 Systematic literature review

As indicated in Section 3.3, based on initial research into the thesis' topic area, there appears to be a trend toward a focus on short-term predictions for safety or navigational purposes. In contrast, this thesis aims at using AIS, and other attributes, for longer-term predictions, or more precisely, port destination predictions. However, because of the exploratory nature of the thesis, the literature review conducted was broad in order to include work that might have taken a different approach to solve the same problem. In order to organize the resulting papers, a categorical separation of papers based on motivation was defined as follows:

0. The paper's motivation deems it completely irrelevant to the topic area.
1. The paper's motivation includes vessel predictions, but on a smaller time or geographical scale making it irrelevant for comparison.
2. The paper's motivation includes destination predictions making it relevant for further analysis.

Category 0 is defined to filter out papers that were irrelevant but could not be excluded by narrowing the search query. In this category, relevance is determined by the studies' topic areas, primary motivators, or methodologies. For example, topics not falling within the maritime topic area have been excluded as they did not appear to include specific insight into the thesis goal which involves specifically analyzing voyage patterns in the shipping industry. Moreover, maritime studies that did not apply any automated or technically applicable solutions were also excluded. Examples of such studies include financial model studies or case studies applied at a specific time interval, vessel segment, or geographical region or country.

Category 1 includes papers that relate to the established trend mentioned earlier where the proposed method seems relevant on a small scale but is ultimately not applicable to the thesis' problem area. Such studies use relevant data sources and technologies but apply them based on different motivations than destination prediction.

Finally, the papers labeled with relevancy 2 falls within *Category 2* and includes papers that fall within the same topic area and are relevant in terms of providing insight into the proposed research questions. In order to determine what papers fitted *category 1* and *2*, papers with a relevance higher than zero were further analyzed in order to determine the following attributes:

- **Motivation** — what was the problem the paper aimed to solve.
- **Objective** — what was the proposed prediction method's objective (e.g. classification, prediction, clustering).
- **Data source** — what data source was used in the proposed solution (e.g. historical AIS data, port data, vessel, or voyage details).
- **Prediction method** — what prediction method was applied to solve the problem.
- **Geographical extent** — to what geographical scopes was the solution applicable.
- **Time interval** — what were the limitations on prediction methods in terms of time or trajectory duration.
- **Validation method** — what methods were applied to validate the resulting prediction method.
- **Validation metrics** — what metrics were used when establishing the validity of the solution.

In this literature review, the following search engines/libraries where used to collect relevant research:

- *Scopus*[1]
- *Oria*[2]
- *ACM Digital Library*[3]

[1] https://www.scopus.com/

[2] http://ntnu.oria.no/

[3] https://dl.acm.org/

These were chosen based on running test queries and evaluating the relevancy and range of the resulting papers. Furthermore, the proposed query had too many Boolean operators for search engines like *ScienceDirect*[4], however, initial testing revealed that there was significant overlap in resulting papers in search engines not used in the study which indicates that the relevant research has indeed been covered.

3.2 Search query and filters

The objective of the literature review was to conduct a broad search detecting papers related to multiple relevant topics such as *vessel destination prediction*, *vessel trajectory prediction*, *vessel availability forecasting*, and *maritime logistics*. Therefore, the search query used in the literature review was designed to find papers within multiple topics and was derived from testing multiple queries on multiple search engines. For instance, the following queries were tested using the search engine provided by *ScienceDirect*:

- “vessel trajectory” OR “ship trajectory” resulted in **421** papers
- ais AND (“vessel trajectory” OR “ship trajectory”) resulted in **150** papers
- ais AND (prediction OR predicting) AND (“vessel trajectory” OR “ship trajectory”) resulted in **108** papers

The above queries returned a large number of papers relevant to *category 1*, so in order to find more relevant papers, more specific queries were also tested:

- “vessel destination” OR “ship destination” OR “vessel availability” resulted in **389** papers
- ais AND (“vessel destination” OR “vessel availability”) resulted in **25** papers.
- ais AND (predicting OR forecasting) AND (“vessel destination” OR “vessel availability” OR “ship supply”) resulted in **18** papers.

Lastly, in order to find detect research approaching the same problem from a different direction such as not using AIS data, the following queries were also tested using *Scopus* because of boolean operator limits on *ScienceDirect*:

- (vessel OR ship OR maritime) AND (destination OR availability OR supply) AND (prediction OR predicting OR forecasting OR logistics) resulted in **894** on *Scopus*
- (vessel OR ship OR “maritime logistics”) AND (destination OR availability OR supply) AND (prediction OR forecasting) resulted in **314** on *Scopus*
- (vessel OR ship OR “maritime logistics”) AND (destination OR availability) AND (predicting OR forecasting) resulted in **92** on *Scopus*

The search terms that seemed to return the most relevant papers were combined into the final query used in the literature review shown in Code listing 3.1.

[4] `https://sciencedirect.com`

Code listing 3.1: Search query used in literature review

```
ais AND (
    predict OR predicting OR forecast OR forecasting
) AND (
    vessel OR ship OR maritime
) AND (
    destination OR availability OR supply OR trajectory OR logistics
)
```

Moreover, the following filters were used to limit the search result:

- The paper must be published in the last five years.
- The paper must be available in English.
- The paper must be available with the publisher subscriptions provided by NTNU.

The search query was limited to a five-year publish interval because there was a drastic decrease in relevant results returned when a ten-year limit was used during preliminary testing. This could be explained by a recent increase in availability and easy access to historical AIS data as well as a recent increase in technological applications within the shipping industry as a whole. However, in an attempt to reduce the number of irrelevant papers, the five-year limit was applied.

The search query was initially executed on the search engine *Scopus*, thus the search query and filters were modified to the search engine's format as shown in Code listing 3.2.

Code listing 3.2: Search query used in Scopus including filters

```
TITLE-ABS-KEY (
    ais AND (
        prediction OR predicting OR forecast OR forecasting
    ) AND (
        vessel OR ship OR maritime
    ) AND (
        destination OR availability OR supply OR trajectory OR logistics
    )
) AND PUBYEAR > 2014 AND (
    LIMIT-TO ( DOCTYPE , "cp" ) OR
    LIMIT-TO ( DOCTYPE , "ar" ) OR
    LIMIT-TO ( DOCTYPE , "re" ) OR
    LIMIT-TO ( DOCTYPE , "ch" ) OR
    LIMIT-TO ( DOCTYPE , "Undefined" )
) AND ( EXCLUDE ( SUBJAREA , "MEDI" ) )
```

3.3 Results

Using the aforementioned (Section 3.1) search libraries, the defined search query returned a total of **109** papers. After removing the overlapping results from the different search engines there were a total of **90** unique papers that formed the foundation of the literature review. First, the papers were evaluated based on the level of relevance as defined in Section 3.1. Out of the **90** papers, **31** fell within *category 0*, **54** within *category 1*, and **5** within *category 2*.

The large number of irrelevant papers resulted from the broadness of the query that was designed to find results in multiple topic areas. Furthermore, there were some papers that were medical in nature but labeled incorrectly in some of the search engines. Usually, these occurred as the term AIS is also an acronym of *Arterial Ischemic Stroke*. Furthermore, some papers that were not publicly available appeared in the search, and other papers were deemed irrelevant as they concerned topics such as mapping fishing areas in a specific region, power and performance predictions using AIS data, or high-level discussions of potential applications of AIS data analysis.

The large number of papers within *category 1* further confirms the general trend of AIS-based predictions as the primary goal of most of the resulting papers were to predict future positions of vessels within shorter time intervals for the purpose of either safety and navigation or anomaly detection. None of the papers within *category 1* seemed applicable to predict vessel destination ports at a global scale, therefore, specific papers within this category are not discussed in detail throughout this section. However, all papers placed within *category 1* are listed in Tables 3.2 to 3.7

The remaining 5 papers (listed in Table 3.1) were deemed relevant enough for further analysis in regards to the proposed research questions listed in Section 1.5. In addition, Lechtenberg et al. 2019, which was discovered during the process of testing queries, was also included in the analysis as it seemed highly relevant toward availability forecasting but did not appear when using the two search engines in the final review.

3.3.1 Research question 1A

What prediction methods can be used to predict vessel destinations?

Firstly, there was a large number of prediction methods that fell within *category 1* that were not directly applicable for destination predictions on a global scale but were applied to smaller-scale positional predictions. These papers usually proposed some manner of clustering algorithm like DBSCAN to classify trajectories and patterns with some form of point-to-point search along trajectories using either a form of Recurrent Neural Network (RNN), Support Vector Machine (SVM), or k-Nearest Neighbor (k-NN) search.

A more relevant approach was proposed in Zhang et al. 2020 that used a Random Forest (RF) -based trajectory similarity measurement combined with frequencies of port visits to predict traveling vessels' next destination port. The DBSCAN algorithm was used to define trajectories by identifying clusters of vessels around port coordinates. These positions were classified as "port-stay points" so that every position between two stay points was part of a vessel trajectory. In the paper, the proposed RF-based approach was compared with several other ML and non-ML methods of predicting destination ports in a similar fashion based on trajectory similarity. Using the proposed RF-based approach, they achieved a "port accuracy" of *66.57%*, and a "city accuracy" of *81.65%*.

Paper	Goal	Pred. method	Geo-extent	Time frame	Validation	Metrics
Karataş et al. 2020	arrival port, arrival time, and next position prediction based on trajectories	LSTM RNN, DBSCAN, K-NN	large, tested regionally	large	10-fold cross validation	accuracy, f1-score, precision, recall
Zhang et al. 2020	Destination prediction based on trajectories and port frequencies	Random forest, DBSCAN	global	any	5-folder cross validation, comparison with other models	port accuracy, city accuracy, MAE, mean distance error
Roşca et al. 2018	predicting arrival times and destinations of vessels	nearest neighbor search on trajectories	large, tested regionally	large	hyperparam. selection by genetic algorithm, train / test data split	general accuracy
Bachar et al. 2018	predicting arrival times and destinations of vessels	venilia based on markov predictive models	large, tested regionally	large	train / test data split (details lacking)	mean distance error for ETA, general accuracy for destination
Andrej Dobrkovic, Iacob, and J. v. Hillegersberg 2018	longer term predictions of vessel arrival times	genetic algorithm clustering	large, tested regionally	large	case study testing, parameter testing, and simulation	general accuracy, extraction quality, execution time

Table 3.1: Papers collected from literature review with relevant geographical and time limitations

Lechtenberg et al. 2019 used an ensemble approach to forecast vessel supply for the dry bulk cargo industry in predefined regions. This implicitly involved predicting port destinations in the form of voyage patterns and extrapolating port availability into regional availability. They mainly used the Markov Decision Process to predict the next destination ports together with XGBoost to predict ETA and anchor time before leaving the current region. The paper claims a *98%* accuracy for regional availability, however, the accuracy for port prediction was not disclosed as well as the size and extent of the predefined regions that presumably would have a massive impact of accuracy as the larger the regions are, the easier it is to predict the next region. This is especially true for smaller vessels that may rarely or never leave a large enough region.

Roşca et al. 2018 and Bachar et al. 2018 were both published as part of *"The DEBS Grand Challenge 2018"* which involved predicting future destinations of vessels given a dataset containing historical AIS data within the Mediterranean sea. Roşca et al. 2018 proposes the nearest neighbor search on coordinates within trajectories where similar points are defined using both distance, speed, and heading. The longest similar sequence, or the longest predicted segment, is used as a basis for querying a vessel's arrival port. Bachar et al. 2018 developed a tool called "Venilia" that uses different ML methods to predict vessels' destinations, mainly, a Markov predictive model. Using their model, they were able to correctly classify 50% of the events from the dataset.

Karataş et al. 2020 proposes and compares a number of different ML-based trajectory similarity approaches including methods proposed in entries to the *"The DEBS Grand Challenge 2018"*. They also used a similar dataset contained with the Mediterranean sea spanning a few months in history. A number of prediction methods trained and evaluated using geographical and navigational parts of historical AIS data. They were further tested using a grid-based mapping of configurable size and resolution. In general, they found that a RF-based model achieved the best results with accuracy for arrival ports around 86%. They also found that a Long-Short Term Memory (LSTM) architecture performed well predicting next positions in a trajectory on a smaller scale.

Andrej Dobrkovic, Iacob, and J. v. Hillegersberg 2018 proposes a genetic algorithm that is trained to cluster waypoints, and use them to establish a directed graph of sea lanes. They discuss the differences and limitations of other clustering algorithms such as DBSCAN which is widely used for trajectory-related clustering. The proposed genetic algorithm can be applied in a similar fashion to the method described Pallotta et al. 2013 by clustering waypoints, detecting sea lanes, and using them to predict vessels' future destination ports. The genetic algorithm is supposedly more suited for clustering in busy areas where individual vessel trajectories are hard to distinguish from others. The suggested approach shows promise toward prediction, however, the main focus of the study is to detect sea lanes while handling missing or varying data. Therefore, the complete implications in regards to destination prediction are unknown on a global scale.

3.3.2 Research question 1B

What information can be used to predict vessel destinations?

From the research within *category 1*, historical AIS data was almost exclusively the only source for predictions. Furthermore, for the most part, only purely spatial attributes of the AIVDM/AIVDO protocol were used. However, when compared to *category 2*, papers in *category 1* were more frequently using navigational information such as COG, SOG, and headings as these features proved crucial to detect collision situations and anomalies in voyage patterns.

All papers listed in *category 2* relied on historical AIS data and only considered the geographical coordinates clustered as trajectories for destination predictions. They all also relied on a varied collection of port data as this is necessary to predict destination ports. Since Roşca et al. 2018 and Bachar et al. 2018 were both published in response to *"The DEBS Grand Challenge 2018"* they relied on the same input data that included AIS and port data from a single region covering the Mediterranean sea. On the other hand, Zhang et al. 2020 had access to 141 million global historical AIS records from 2011 to 2017 and a global port database consisting of over 10 000 ports.

3.3.3 Research question 1C

To what extent do methods proposed in existing work vary in scope of applicability?

As mentioned frequently in this section, all papers from *category 1* were, in some manner, limited by either geographical extent or time intervals as global destination prediction was not the focus of these papers. There were slight variances in these limitations ranging from time limitations of minutes to hours with some papers considering predictions up to one day. However, since a voyage can last longer than one month in many cases, positional predictions accurate up to one day in the future do not seem applicable to this thesis' problem area.

The papers within *category 1*, were less limited by time and geographical extent. For instance, ibid. and Lechtenberg et al. 2019 both were completely unrestricted by geographical extent or time frames, however, the first did require a current traveling trajectory to predict the next destination port but was capable of making predictions for any voyage no matter the length or duration.

However, the solutions proposed from the remaining papers listed in Table 3.1 were only applied, or tested, regionally. Roşca et al. 2018 and Bachar et al. 2018 proposed solutions to the same challenge using the same dataset containing data within the Mediterranean sea, however, it seems as both approaches are unrestricted in terms of time limitations. Karataş et al. 2020 also proposed an apparent general destination prediction method, however, they were also limited by a dataset only covering the Mediterranean sea and only containing records spanning a couple of months. Additionally, Andrej Dobrkovic, Iacob, and J. v. Hillegersberg 2018's proposed solution to long-term predictions was also only validated on limited regions which were two Dutch provinces.

In conclusion, even papers that set out for long-term destination predictions are mostly all focused on a particular area or region, and although the proposed solutions seem promising, they do not fill the global requirements of this thesis' goals. From the related work, it is apparent that only the solutions proposed in Lechtenberg et al. 2019 and Zhang et al. 2020 are globally extensive.

3.3.4 Research question 1d

How can the validity of predictions made based on different prediction methods be established?

From *category 1*, there was a multitude of different validation approaches that were relevant to smaller-scale predictions. In these methods, the most prevalent metrics used were distance-based error rates as the positional accuracy of the models is important for topics such as collision detection. Other standard ML related validity metrics also occurred such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), F1-score, and general prediction, or classification, accuracy.

Lechtenberg et al. 2019 does not disclose the evaluation process in much detail, however, it is mentioned that there is a standard division of 90% training data and 10% test data. It mentions using both MAE and RMSE as metrics measuring the quality of their approach which are both frequently used metrics ML.

Zhang et al. 2020 describes a more thorough evaluation process using 5-folder cross-validation which is a well-established process to ensure a model's accuracy is reliable and not a result of overfitting. The proposed model was compared under similar conditions with several other trajectory similarity measurements such as the aforementioned SSPD algorithm but also a few ML-based approaches. The metrics used to indicate validity were mainly Average Prediction Distance Error (APDE) based on the distance from the predicted port to the actual arrival port, port accuracy which was also extrapolated to city accuracy.

Karataş et al. 2020 employed a 10-folder cross-validation process to validate their approach which was compared to a number of different ML-based prediction models. General prediction accuracy was the main validity metric in addition to other standard metrics, namely F1-score, precision, and recall.

Neither Bachar et al. 2018 nor Roşca et al. 2018 does not describe an evaluation process in high detail. However, in ibid., there are mentions of testing two different data structures and their impacts on runtime performance as well as a general best score of 0.8249 for predicting arrival ports and a mean error rate for ETA predictions.

Finally, Andrej Dobrkovic, Iacob, and J. v. Hillegersberg 2018 is harder to compare to as destination predictions were not directly applied and tested in the study. In regards to route construction and pattern extraction, they ran tests in a simulated environment in order to establish the accuracy of their genetic algorithm that tries to establish sea lanes and routes. They reported an accuracy of 87.5% in one simulation case and a lower score of 75% in a scenario including missing parts in the input data which the algorithm is trained to handle.

3.3.5 Research question 1 – summary

How can AIS data combined with specific vessel details be applied to predict future destinations of maritime vessels?

As discussed throughout this section, the existing literature within the thesis' topic area seems to exhibit a few trends. First, the majority of the research discovered mostly focused on short-term navigational predictions of vessels, usually for the purpose of collision avoidance, anomaly detection, or safety and management within ports or smaller regions. Second, there is a very limited amount of research that focuses on longer time intervals or geographical extent, and these studies are also quite limited in the sense that they rarely consider global port destination predictions, but rather port destinations within a given region. Lastly, there was no research found that considered much more than purely geographical attributes of historical AIS data. When only considering geographical travel patterns of vessels, it is implicitly assumed that all vessels behave in a similar fashion which, as shown in Figure 2.6, is not the case. This is especially true for highly specialized vessels such as service vessels for oil platforms, or passenger vessels that are not at all comparable to dry bulk or tanker vessels. Since the existing literature is also lacking in general global destination prediction methods that consider specific details for individual vessels, it can be concluded that the existing literature does not provide enough insight into RQ1 as a whole.

3.3.6 Research question 2

What is the impact of vessel segmentation by type, size, or capacity on prediction methods, or vessels' general predictability?

The related work within this field seems to be substantially limited in terms of considering different vessel types or segmentation. None of the discovered studies that apply a general methodology to predict any vessels' future destination or trajectory consider the differences in vessel size, capacity, or type. Some studies conducted research on forecasting within one specific segment or sub-segment. Lechtenberg et al. 2019 in particular, developed a prediction method for dry bulk cargo vessels. Their developed prediction method achieved a high level of accuracy when predicting the future destination region of dry bulk vessels. However, since they did not apply their method on other types of vessels, or analyze the prediction results per sub-segment within the dry bulk segment, the study does not provide insight into the impact of segmentation or correlations between vessel size and predictability. Thus, the existing literature does not provide enough insight into RQ2 or any of the sub-questions within it.

Paper	Motivation	Pred. method	Geo-extent	Time scale	Validation	Metrics
Braca et al. 2018	detecting vessels turning off AIS transmitters	Ornstein-Uhlenbeck (OU) mean-reverting stochastic process	large	medium to large	case study using scenario/simulation	accuracy, use-case effectiveness
Konstantinos et al. 2018	detecting search and rescue activity before they are emitted	random forest	regional	N/A	cross validation folder	F1 score, recall, precision
Andrej Dobrkovic, Iacob, and J. V. Hillegersberg 2015	predicting arrivals for Dutch logistics service providers (LSPs)	directed graph, DBSCAN, genetic algorithm	large	24 hours	simulation	extraction quality, efficiency, noise tolerance
El Mekkaoui et al. 2020	ETA prediction for known destinations	FFNN, RNN, LSTM, GRU	regional	large	cross validation	MAE, MSE
Jia et al. 2019	destination classification for Latin-American crude oil exports	random forest	large	cross validation	Gaussian naive Bayes (GNB) classification	accuracy
Jung et al. 2019	trajectory anomaly detection (ACM DEBS GC 2018 challenge)	Hausdorff distance	region	medium	predictability increase after anomaly removals	accuracy, model performance increase
Lei 2020	constructing a database for detecting collision situations	time/distance at closest point (DCPA/TCPA), clustering	regional	hourly	region analysis, number and distribution of conflicting trajectories	N/A
Ma et al. 2020	collision avoidance	genetic optimization algorithm, particle swarm optimization, neural network	small	small	cross validation	MSE

Table 3.2: Papers gathered from literature study labeled with relevancy level 1 whose objective was classification (part 1/2).

Murray and Lokukaluge P. Perera 2020	collision avoidance, anomaly detection	GMM	regional	N/A	top/bottom eigenvector analysis, Mahalanobis distance analysis	accuracy
Nguyen et al. 2018	arrival port and ETA classification	sequence to sequence on a grid	regional	medium	case study	N/A
Patmanidis et al. 2016	anomaly detection and route prediction	linear filtering using ARMA models	region	small	simulation	distance error
Prochazka and Adland 2020	feature selection for enhancing predictions	N/A	regional	N/A	N/A	N/A
Rong et al. 2020	collision probability prediction	Gaussian process	small	small	case study	reliability / accuracy
Shen et al. 2020	improving detection of fishing activities	RNN, LSTM	regional	N/A	cross validation	F1 score, AUC, accuracy
Tang, Wei, et al. 2020	anomaly detection and safety in maritime navigation	DBSCAN, mesh-based deviation detection on directed graph	regional	N/A	simulation	N/A
Watawana and Caldera 2018	collision detection	SVM, Naive Bayes, CART	regional	small	case study	prediction accuracy
Wen et al. 2020	defining routes between connected regions for route planning	DBSCAN, ANN	global	N/A	N/A	N/A

Table 3.3: Papers gathered from literature study labeled with relevancy level 1 whose objective was classification (part 2/2).

Paper	**Motivation**	**Pred. method**	**Geo-extent**	**Time scale**	**Validation**	**Metrics**
Hamada et al. 2021	Main: effective information from marine big data + predicting the demand of ships	"deep learning, clustering"	global	any	25/75 validation/training	"accuracy, standard deviation"
T. Wang et al. 2021	Improved trajectory clustering for prediction and anomaly detection	convolutional auto-encoders + clustering MFA+K-means	tested on a single city area	small	"not explained, compared to other methods"	"precision, accuracy, recall, f1 score"
Alexander Dobrkovic et al. 2015	systematic mapping of short to long term predictions	literature review (N/A)	N/A	N/A	N/A	N/A

Table 3.4: Papers gathered from literature study labeled with relevancy level 1 whose objective was neither classification nor prediction.

Paper	Motivation	Pred. method	Geo-extent	Time scale	Validation	Metrics
Alizadeh et al. 2020	trajectory similarity to predict positions at time intervals	spatial distance, bi-directional distance, speed distance	region	10, 20, 30 min.	case study	accuracy, SSI
Alizadeh et al. 2021	Vessel trajectory prediction for collision avoidance	LSTM (RNN) and trajectory similarity	region	10-40 min.	cross validation	distance accuracy
Borkowski 2017	collision avoidance integrated in navigation systems	ANN, data fusion, GRNN	meters	minutes	integrated and tested in real navigational system	RMSE
Brandt and Grawunder 2017	moving objects prediction	moving object data stream management systems, kNN	region	10 min.	test cases	not explained
Burger et al. 2020	filling in gaps in AIS data	discrete Kalman filters, LRM	small	small	single cases analysis	distance error
Chen et al. 2020	cluster reconstruction for short time frames	NPC clustering finding best possible next points	small	small	extensive comparisons with other methods	accuracy, distance error
Dalsnes et al. 2018	collision avoidance for autonomous ships	NCDM	meters	minutes	cross validation	RMSE
Dijt and Mettes 2020	collision avoidance for autonomous ships	sequence to sequence neural network	meters	minutes	cross validation	RMSE, MAE
Ding et al. 2020	longer time trajectory predictions	LSTM	meters	5-20 min.	training / validation sets (8:1)	MSE
Forti et al. 2020	sequence-to-sequence RNN approach	RNN, LSTM encoder-decoder	region	small	5-fold cross validation	RMSE
Guo et al. 2018	regional trajectory predictions	K-order multivariate Markov chain	region	small	simulation, experiments	accuracy
Hexeberg et al. 2017	collision detection	single neighbor search	region	10 min.	cross validation, selected scenarios	RMSE
Jin et al. 2020	predictions for collision	RNN, LSTM	region	small	model simulation	distance error, MAE, SSE

Table 3.5: Papers gathered from literature study labeled with relevancy level 1 whose objective was prediction (part 1/3).

Kim and Lee 2018	predictions for Vessel Traffic Service (VTS)	NN	small	small	case study on region	speed and distance error
J. Li et al. 2018	data mining for predictions	RBF RNN	small	hours	case study on river in China	trajectory accuracy
W. Li et al. 2019	collision avoidance	LSTM, LCS, DBSCAN	meters	15 min.	case studies	distance error
Lian et al. 2019	particle filtering, least squares estimation	linear prediction, least squares, particle filtering	meters	minutes	simulations	distance error, speed error
J. Liu et al. 2019	real-time predictions at sea	SVR, ACDE, RNN	small	small	cross validation	distance error
X. Liu et al. 2020	predictions for ship management, fill in missing AIS data	LSS-VM, cubic spline interpolation	small	small	case studies	trajectory accuracy
Mao et al. 2018	database for trajectory prediction and mining	trajectory interpolation, grid search, SLFN	region	20-40 min.	case studies	distance error
Murray and Lokukaluge Prasad Perera 2018	collision detection for autonomous vessels	nearest neighbor search	meters	5-30 min.	cross validation	RMSE
Murray and Lokukaluge P. Perera 2019	collision avoidance	Gaussian mixture modelling, PCA	small	small	case studies	distance error
Murray and Lokukaluge Prasad Perera 2020	trajectory predictions for early warnings and safety	GMM clustering, dual auto-encoder	regional	30 min.	case studies	accuracy at time intervals
Rong et al. 2019	modelling uncertainty of trajectory predictions	Bayesian model, Gaussian Process	small	10-30 min.	case study in region, training / validation data	accuracy, distance error
Suo et al. 2020	trajectory predictions for early warnings and safety	GRU (gate recurrent unit), DBSCAN, comp. with LSTM	tested on single port in china	small, minutes to hour	training, validation, test set (not defined how much)	accuracy

Table 3.6: Papers gathered from literature study labeled with relevancy level 1 whose objective was prediction (part 2/3).

Tafa et al. 2019	synthetic route representation and predictions	DBSCAN, route similarity probability model	regional	10-80 min.	simulation	accuracy
Tang, Yin, et al. 2019	collision avoidance for autonomous ships	LSTM	region in china	20 min.	cross validation	MAE, MSE
Uney et al. 2019	forecasting trajectories from historical and streaming trajectories	directed grid based Bayesian model / Gaussian mixture forecast density	grid-based region	N/A	regional case study	not explained
Virjonen et al. 2018	predictions for port management in Finland	k-NN	regional	several hours	leave-one-out cross validation (LOOCV)	distance accuracy
C. Wang et al. 2020	predicting vessel berthing trajectory for safety and collision avoidance	Bi-GRU (tensorflow, keras)	port-based	small	cross validation	MSE
Xiao et al. 2020	collision avoidance, effective queries, more effective predictions	knowledge based particle filtering (PF), MLNN	small	3-10 min	case studies	sog, coc, and distance error
You et al. 2020	sequence-to-sequence RNN approach	seq2se1 GRU, RNN, encoder/decoder	small, limited to 10m trajectories	10 minutes	case studies, cross validation	AdaGrad, RMSProp
Zheng et al. 2020	combining multiple data sources like GPS and ARPA with AIS to improve predictions for safety	LSTM (on different data and a fusion component to merge the predictions)	small	small	cross validation	MSE
Zhou et al. 2019	collision avoidance in busy areas	back propagation neural network	region (area in china)	small	cross validation	RMSE

Table 3.7: Papers gathered from literature study labeled with relevancy level 1 whose objective was prediction (part 3/3).

Chapter 4

Methodology

In this chapter, the methodology of the proposed solution is explained in detail. This section is divided into sequential subsections that each describe a step in the process used to arrive at the proposed dataset, the formulation of the analytical problem to be solved, and the Machine Learning (ML) related data preparation, training, and evaluation.

4.1 General approach overview

Based on the findings from the literature review conducted in Section 3.1, it is clear that the existing work is limited in terms of a general and global prediction solution for vessel destination ports. Thus, this thesis proposes a method that is able to predict vessels' future destination ports using a combination of positional data from the AIVDM/AIVDO protocol and vessel segmentation values. This is an important objective of the thesis as no related studies seem to take additional vessel information into account. The proposed solution is not restricted by specific geographical regions nor time intervals and should form a foundation of which it is possible to extend with more features, or data attributes, regarding the traveling vessels and voyages. The method of developing the proposed solution can be divided into the following steps:

1. Construct voyages and trajectories using a voyage definition derived from the departure and arrival detection described in Section 2.3.1.
2. Sample, or simplify, the trajectories to make them more comparable using vessel similarity measurement methods.
3. Calculate the Most Similar Trajectory's Destination (MSTD) and the similarity value for every voyage's trajectory.
4. Collect the historical data attributes to be used for Machine Learning (ML) including departure and arrival ports, vessel segmentation values, MSTD values, and trajectory lengths.
5. Train a ML model to predict the arrival ports of voyages using the dataset constructed.

4.2 The initial data processing

This section describes the initial dataset used in the proposed solution which is later processed and used to train a ML model for predictions. This data foundation is provided to the author by Maritime Optima AS (MO). Moreover, for this thesis, data used in the analysis is stored in a separate, dedicated PostgreSQL database also hosted in MO's cloud computing environment.

4.2.1 Positional historical AIS data

The first step in the dataset processing is to collect a historical set of AIS data. In this thesis, this data provided by MO contains more than 1.5 billion positional records for over 65 000 unique vessels starting from December 2019 and is continuously collected. In this thesis, circa 1.2 billion records ranging from December 2019 to March 2021 were used for the proposed solution. The historical records were copied in batches from MO's database into a separate database used in this thesis in a table called *vessel positions history*. This table contains the following relevant attributes for each historical record:

- id – a sequential identifier
- imo – the IMO number of the vessel that transmitted the position.
- mmsi – the MMSI number of the vessel that transmitted the position.
- position – a geographical coordinate of the vessel in the *Mercator* projection.
- timestamp – the UNIX timestamp (seconds since Unix Epoch) of when the position was transmitted by the vessel.

In the process of copying data to the dedicated database, each position's coordinate is validated by ensuring that it follows the bounds of its projection, i.e., that the longitude value is between -180, and 180 degrees and the latitude is between -90, and 90 degrees. If a coordinate has invalid values, it is disregarded. Furthermore, positions that lie exactly on the north and south bounds, or exactly at coordinates *(180, 90)* and *(-180, -90)* are also disregarded as these positions are impossible places to navigate but are still frequently seen in the database. Figure 2.5 in Section 2.2.3 shows a visualization of an extract of 200 million records from the historical AIS database which shows the extent of the collected positions.

Furthermore, as also mentioned in Section 2.2.3, IMO numbers and MMSI numbers are divided up in the positional and static AIS reports. Therefore, MMSI numbers in positional data must be matched to IMO numbers in the static information (which contains both) to collect both identifiers in the historical AIS database. The IMO identifier is required to extract information such as vessel segments and sub-segments as these are initially constructed using information from static records. Positions transmitted by a MMSI number that does not map to a known IMO number, or have invalid values for either, are disregarded. The validity of both values can be determined following the AIVDM/AIVDO protocol which defines how these numbers are constructed and used.

4.2.2 Segments

As described in Section 2.3.2, vessel segmentation values are additional attributes that indicate a vessel's type, dimensions, and capacity. These labels are thought to provide insight into the traveling patterns of vessels. Thus, this information is important for this thesis's proposed solution. MO has vessel segmentation information for every unique vessel collected by AIS data. This information is collected and stored in the dedicated database in a table called *vessel segments*. This table contains information per vessel and has the following relevant attributes:

- imo – the IMO number of the vessel.
- segment – the vessel's segment value, e.g. *dry bulk*, *tanker*, *chemical*, etc…
- sub-segment – the vessel's sub-segment value, e.g., *mini bulker*, *handysize*, *Panamax*, etc…

Finally, it is worth noting that some vessels can function as two different types of vessels such as tanker vessels that also function as chemical transport vessels. These "combo" vessels contain multiple entries in the segmentation database table for each of the functions it serves. However, they also contain a dedicated entry where the segment value is "combo" which can have a specific range of sub-segments. For technical reasons, it is more practical to assume that every vessel only has one segment and one sub-segment, therefore, for combo vessels, only the combo segment itself and its sub-segment are considered.

4.2.3 Ports

Next, the traveling vessel's departure and arrival port are required to predict vessels' future destinations, as destinations are defined ports. As already described in Section 2.3.3, MO has a large number of ports available in a port database out of which around 5600 are considered relevant for the shipping industry. For this thesis, only these 5600 relevant ports are considered for the analysis, thus, these are also stored in the dedicated database in a table called "ports". This table contains the following relevant attributes:

- locode – the port's unique identifier following the UN/LOCODE protocol.
- position – the port's geographical coordinates specified in the Mercator projection.
- name – a text value for the name of the port.

4.2.4 Vessel transitions

As described in Section 2.3.1, vessel transitions are historical events where a vessel's AIS navigational status transitions from a status indicating that it is moving to the status "MOORED" and vice versa. These events are mapped geographically to the closest known port within a 25-kilometer radius, thus, vessel transitions provide a historical record of port arrivals and departures. MO has more information available in their transition data, however, only vessel arrival and departure data are of relevance to the proposed solution. The relevant data is stored in the dedicated database as a table called *vessel transitions*. This table contains vessel identifiers, the event's mapped port, the Unix timestamp of when the event occurred, and the transition type indicating whether the vessel arrived or departed. Table 4.1 shows an example extract from the transitions data for a single vessel that used the navigational status correctly. It shows that when sorted by time, the events follow a pattern of sequential arrival and departures from different ports.

IMO	MMSI	Transition	Timestamp	Port Code
9824083	538008866	ARRIVAL	1595383670	KRONS
9824083	538008866	DEPARTURE	1596177702	KRONS
9824083	538008866	ARRIVAL	1599869735	BRITQ
9824083	538008866	DEPARTURE	1600002777	BRITQ
9824083	538008866	ARRIVAL	1603942962	CNZNG
9824083	538008866	DEPARTURE	1604191770	CNZNG

Table 4.1: Example rows for a single vessel in the vessel transitions table

4.3 Vessel voyage definition

After the initial data foundation is constructed as described in the previous section, the next step is to construct voyages and voyage trajectories based on the historical AIS data. As described in Section 2.1.1, throughout this thesis, a vessel voyage has been defined based on vessel transitions derived from AIS navigational statuses. This is mainly because it is thought to provide more valuable predictions for the people working in the industry. This hypothesis was confirmed by the collaborative company Maritime Optima AS (MO) but has also been corroborated by interviewing shipping experts as will be later discussed in Chapter 5. This section describes the methods tested and used to construct voyages from the initial data foundation described up to this point in Chapter 4.

4.3.1 Cluster-based voyages

As discussed in Section 2.1.1 and as shown in the literature study conducted in Section 3.1, one alternative vessel definition is based on detecting stopping motions close to ports in AIS messages. This approach to voyage definitions only considers geographical, or navigational, information in the AIVDM/AIVDO protocol to look for recurring patterns that identify a vessel as stopped close to ports. An existing example using this approach for voyage destination prediction was used in the study presented in Zhang et al. 2020. They used the Density-based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm to find clusters of AIS positions close to ports and defined a voyage as positions transmitted between two clusters for a vessel. In order to investigate the ramifications of using such an approach, a similar method was developed in this thesis as well.

First, positional AIS records were fetched per vessel as well as the positions of every port in the database. Next, the vessel positions were parsed and passed to the DBSCAN algorithm which detects clusters of positions based on the parameters *epsilon*, and *minimum samples*. The minimum samples parameter defines the minimum amount of positions required for a cluster to form while the epsilon defines the minimum distance between two points required for points to be added to a cluster. Since the algorithm is unsupervised, there is no real automatic method of finding the best parameters, therefore, the parameters must be chosen by using manual inspection. In this approach, the parameters used in Zhang et al. 2020 were used as a starting point and then changed until fitting parameters were found. The epsilon parameter used in the aforementioned paper was appropriate, however, the minimum samples parameter required a higher value. Thus, the parameters *epsilon* and *minimum samples* were set to 1.095×10^{-5} and 5 respectively. After DBSCAN was used to cluster records, each resulting cluster was mapped to its closest port by measuring the distance between the cluster's center-most point to each port. If the closest port was further away than a given threshold set to 1km, the cluster was disregarded. Code listing 4.1 shows an extract of Python code used to cluster vessel positions and map them to their closest port.

Code listing 4.1: Python code used to cluster AIS records using DBSCAN and map the clusters to their closest ports.

```python
def dbscan_reduce_vessels(df, port_df,
                          epsilon=0.00001095,
                          min_samples=5,
                          algorithm="ball_tree", metric="haversine"):
    """ reduce vessel positions to clusters using DBSCAN

    returns clusters and each cluster's closest port in df
    """

    # dbscan_reduce wraps sklearn.cluster.DBSCAN and returns df with clusters
    clusters = dbscan_reduce(df, epsilon, min_samples, algorithm, metric)

    data = []
    for cluster in clusters:
        (lat, lon) = get_centermost_point(cluster)
        # find closest port.
        # skip if closest is further away than threshold = 1000m
        closest_port = get_closest_port(port_df, [lat, lon], 1000)
        if closest_port is None:
            continue

        timestamp = get_latest_cluster_timestamp(df, cluster)
        (imo, mmsi) = get_properties_at(df, lat, lon, ["imo", "mmsi"])
        data.append({
            "lat": lat,
            "lon": lon,
            "closest_port": closest_port,
            "timestamp": timestamp,
            "imo": imo,
            "mmsi": mmsi,
        })

    return pd.DataFrame(data)
```

Figure 4.1 shows an example of positional records being clustered and mapped to ports. In this example, the vessel called *Jonas Oldendorff* travels from Mauritania, stops in Morocco, and finally stops in France. The detected clusters in the vessel's positions closely relate to the three ports, however, in other cases, vessels were found to stop further away from ports. In these cases, the vessel most likely anchored outside of an area while waiting to arrive at multiple ports. In these cases it is correct to not consider those clusters as arrivals, however, it is impossible to know the vessel's intent when it is stopping without considering additional information such as the port's capabilities, restrictions, or the vessel's navigational statuses.

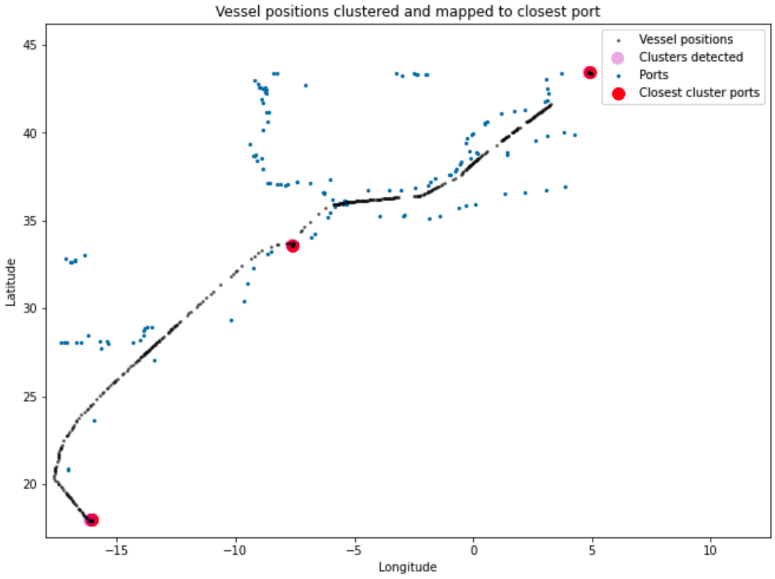


Figure 4.1: Example of DBSCAN clustering of AIS records where clusters are mapped to the closest available port.

Finally, given a number of clusters that were close enough to map to ports, each position transmitted by the vessel between two clusters was stored as the vessel's trajectories. Figure 4.2 shows the same example from Figure 4.1 but with the resulting trajectories constructed using the clusters.

The resulting voyages shown in Figure 4.2 seemed promising, however, for other vessels the quality of the extracted voyages was not consistent. For example, small vessels traveling very short distances such as passenger vessels transmitted positional records at a much higher density than larger vessels traveling further, thus, the chosen parameters did not perfectly fit these vessels. Moreover,

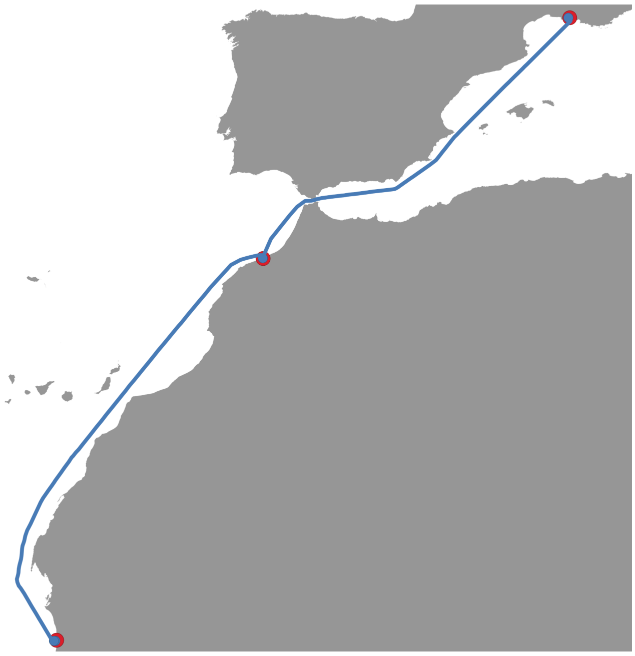


Figure 4.2: Two subsequent voyages derived from DBSCAN clustering and mapping to existing ports. The blue dots are detected clusters, the red ones are the closest mapped port, and the line represents the trajectory of connected AIS records.

for larger vessels, clusters were often detected mid-voyage whenever the vessel stopped for longer periods of time such as when bunkering (refueling) during a voyage. Although this was expected, it is assumed that ignoring shorter stops during voyages in order to predict the vessels' final destinations has higher commercial value when compared to often predicting the next bunkering port as the next destination for many vessels. Lastly, although the DBSCAN method's processing performance was quite efficient, additional processing such as mapping each detected cluster to the closest port was less so. The aforementioned factors made the final cluster-based voyage solution impractical when compared to the alternative described in Section 2.1.1 and subsequently in Section 4.3.2 which considers the vessels' navigational statuses. Therefore, the cluster-based approach was abandoned in favor of the alternative approach which proved to be more practical in implementation and potentially more valuable for commercial actors. Section 4.3.2 describes how this alternate approach was implemented as well as examples highlighting the aforementioned benefits of the alternate voyage definition.

4.3.2 Transition voyages

As vessel transitions provide a historical record per vessel of arrivals and departures based on AIS statuses it can be used to derive vessel voyages. Given two entries in the vessel transitions table for a single vessel, sorted by time, where the first is a departure from a port, and the second is arrival at another port, a voyage can be defined as starting at the timestamp of the first departure transition and ending at the subsequent arrival transition. For example, in Table 4.1, there are six transition events for a single vessel ordered by time. Based on these events, two different voyages can be defined:

1. Vessel departed port *KRONS* on the 31st of July 2020 and arrived at port *BRITQ* on the 12th of September 2020.
2. Vessel departed port *BRITQ* on the 13th of September 2020 and arrived at port *CNZNG* on the 29th of October 2020.

Using the vessel transition information, the first step is to deduce voyages based on sequential arrival and departures for vessels. After a voyage has been defined, the second step is to find the trajectory of the traveling vessel. This can be deduced from the vessel positions history table as every AIS positional report transmitted between the derived departure and arrival timestamps from the traveling vessel forms a geographical trajectory from the departure and arrival port.

Constructing voyages

Constructing voyages based on vessel transitions and positional records can be summarized in the following steps:

1. Extract vessel transitions per vessel ordered by time.
2. Define voyages based on subsequent departures and arrivals from the vessel transition data (Code listing 4.2).
3. For each voyage, extract every positional record between vessels' departure and arrival timestamps sorted by time.
4. Validate the geographical trajectory including applying a “noise filter” (Figure 4.4).
5. Store voyages with validated trajectories in the transition voyages table.

1. Extracting vessel transitions First, vessel transitions were simply fetched from the database ordered by vessel identifiers and timestamps. In order to make the voyage building process idempotent, the last voyage constructed per vessel was pre-fetched from the voyage table, and only transitions for a given vessel that occurred after the latest constructed voyage were included in the next steps. In this way, the voyage builder process could run multiple times when new data became available without affecting existing data.

2. Define voyages based on subsequent departures and arrivals This next step was a relatively straightforward process consisting of looking for transitions following the pattern of departures immediately followed by an arrival at a different port. The algorithm used to compute voyages based on a vessel's transition events is shown in Code listing 4.2.

Code listing 4.2: Golang code used to compute voyage times from vessel transitions. The code has been reduced slightly for readability.

```go
func getVoyageTimes(transitions []VesselTransition) []Voyage {
        var voyages []Voyage
        j := 0
        for i, current := range transitions {
                // start at the first departure event
                if current.Transition != "DEPARTURE" {
                        continue
                // get the subsequent transition
                j = i + 1
                if j >= len(transitions) {
                        break
                }
                next := transitions[j]

                // ensure the next event is an arrival at a different port
                if next.Transition != "ARRIVAL" {
                        continue
                }
                if current.PortCode == next.PortCode {
                        continue
                }

                voyages = append(voyages, Voyage{
                        IMO:                current.IMO,
                        MMSI:               current.MMSI,
                        DeparturePort:      current.PortCode,
                        DepartureTimestamp: current.Timestamp,
                        ArrivalPort:        next.PortCode,
                        ArrivalTimestamp:   next.Timestamp,
                })
        }
        return voyages
}
```

3. Construct trajectories from the defined voyage times Next, the voyage times computed were used to construct geographical trajectories. This process is, in essence, a simple matter of extracting positional records from the vessel positions history table for the given vessels between the departure and arrival times ordered by time. However, some trajectories were found to be invalid or contained abnormal patterns, therefore, all trajectories were also validated based on coherence and distance between two points in a trajectory.

4. Trajectory validation During the process of building trajectories, several trajectories were discovered that showed peculiar shapes. For instance, there could be large gaps in certain parts of the trajectory or fluctuations when the vessel was stationary. Therefore, a validation step was added to the voyage construction process, for instance, if the distance between two points is sufficiently large, there is most likely missing coverage in the AIS data and the trajectory should probably be skipped. Furthermore, another issue was detected where some vessels showed a seemingly coherent trajectory except for a section of it where the longitude or latitude value fluctuated with extreme distances. This often happened when a vessel was stationary or moving slowly and it transmitted many positions in close proximity.

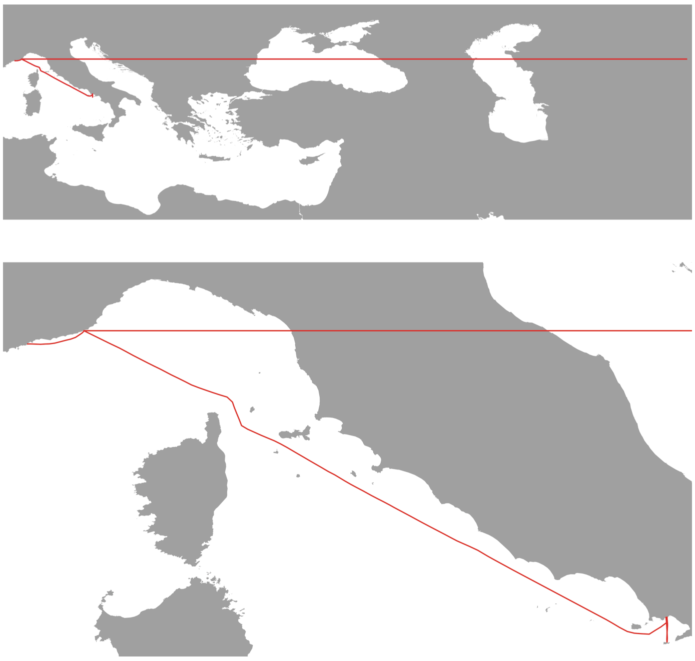


Figure 4.3: Example showing a "noisy" trajectory presumably caused by GPS inaccuracy or equipment error

An example of this issue is visualized in Figure 4.3 which shows a voyage starting in Monaco and ending in Naples, Italy. During a stopping point in northern Italy, the vessel transmitted two longitude values placing the vessel in the middle-east while then continuing the journey arriving in Naples, Italy. This issue is presumably caused by issues with the GPS signals sent by the AIS transmitter onboard the vessel or by some other equipment error. Excluding the fluctuated segment of the trajectory, the remaining trajectory is completely valid, thus, if it is possible to remove the invalid part of the trajectory, the remainder could be further used in the analysis. Therefore, a "noise filter" was employed to detect and cut away fluctuations in otherwise valid trajectories.

The noise filtering employed in the trajectory builder is shown in Figure 4.4 where the red segment fluctuates in an otherwise valid trajectory and is therefore excluded from the remaining trajectory. For every point in the trajectory, the algorithm checks the distance between the current and the next point as well as the time difference between the two points. Using the distances in space and time, it calculates the speed the vessel would require to travel from the first to the second point. If the speed required was more than 50 knots, the segment was

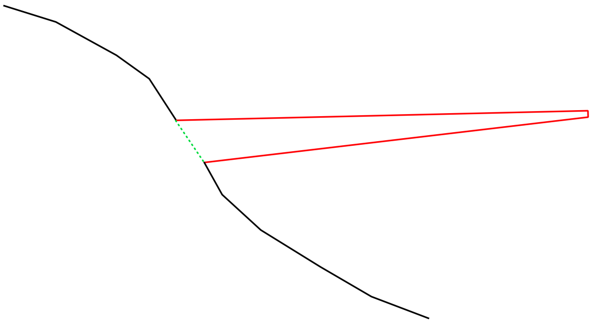


Figure 4.4: Noise filter algorithm cutting out points in a trajectory detected as noise. The red segment is cut out and the black segments are tied together as shown with the green dotted line.

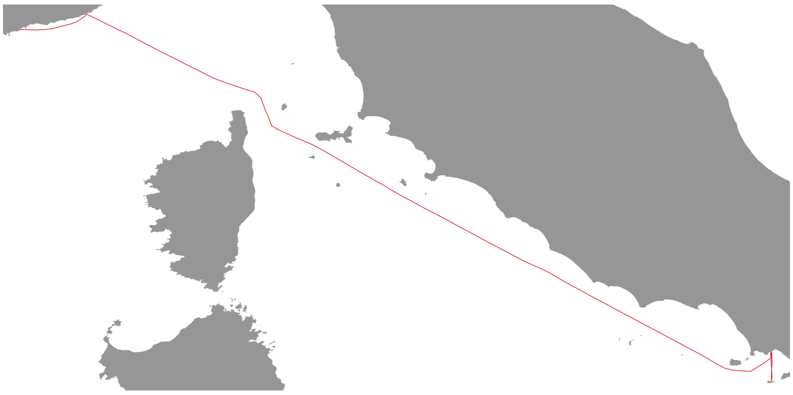


Figure 4.5: The example trajectory shown Figure 4.4 from Monaco to Naples after noise filtering.

invalid. The next point is then compared to the first to see if there is a possible valid path to the third point. If there is, the second point is disregarded from the trajectory. The algorithm is given a tolerance of four invalid points before it disregards the entire trajectory. Code listing 4.3 shows the function used to find the next valid point in a trajectory, if it returns an error, the trajectory is disregarded. Figure 4.5 shows the same trajectory from Figure 4.3 after the following algorithm has filtered out fluctuating segments.

Code listing 4.3: Golang code used find the next valid point for any given point in a trajectory.

```go
// nextValidPoint returns the index of the next valid point checking distances to
// every point within tolerance. If no valid distances were found wihin tolerance,
// it returns an error. If the last point in trajectory was reached, -1 is returned
func nextValidPoint(start, tolerance int, positions []VesselPosition) (int, error) {
        a := positions[start]
        for j := start + 1; j <= start+tolerance; j++ {
                if j >= len(positions) {
                        return -1, nil
                }

                n := positions[j]
                dist := DistanceHaversine(Point{a.Lon, a.Lat}, Point{n.Lon, n.Lat})
                // use the absolute value in case the trajecotory is not sorted
                timeDiff := math.Abs(float64(n.Timestamp - a.Timestamp))

                // calculate the required speed to reach the given point with the
                // given time difference * 1.94385 to konvert m/s to knots
                requiredSpeed := (dist / timeDiff) * 1.94385
                // if required speed was >= 50kt, move on to next point
                if requiredSpeed < 50.0 {
                        return j, nil
                }
        }
        // no reasonable distances were found within tolerance
        return -1, errors.New("trajectory segment too noisy")
}
```

5. Store voyages with validated trajectories in the transition voyages table Finally, when the voyage trajectories have been constructed and validated, they are collected in a database table called transition voyages which contains the following relevant attributes:

- imo – an identifier for the vessel.
- mmsi – an identifier for the vessel.
- departure port – voyage departure port's locode.
- departure timestamp – the time of departure.
- arrival port – voyage arrival port's locode.
- arrival timestamp – the time of arrival.
- trajectory – 3D linestring with longitude, latitude, and timestamp for each point.

It is worth noting that the trajectories are stored as 3D PostGIS linestring geometries where each point contains an x, y, and z value where the z value holds the UNIX timestamp of the positional record. Keeping the timestamp value is necessary for sampling trajectories based on time, and keeping the time values stored directly in the trajectory geometry saves an extra table for trajectory points. Thus, storage complexity is somewhat reduced while retrieval complexity remains the same as with a separate trajectory point table. This is becuase the PostGIS function *ST_DumpPoints* can be used to retrieve the *XYZ* coordinates similarly to that of a *JOIN* operation on a separate table. The performance benefits are negligible.

4.4 Data processing for Machine Learning (ML)

After the initial data set has been collected and vessel voyages have been defined and constructed, the next step is to build the final training dataset to be used for analysis and Machine Learning (ML). This section describes every step in the process used to construct this dataset based on data described up to this point in this chapter.

4.4.1 Trajectory sampling

Vessels transmit AIS records at different frequencies and messages collected via satellite are collected at different frequencies as the satellites have different orbits. Therefore the frequency, or density, or records in trajectories can not be expected to be standardized. Furthermore, as vessels travel at different speeds, two trajectories with similar start and end positions might have different shapes and contain a different number of points. In addition, as discussed in Section 2.1.1, one disadvantage of relying on AIS navigational statuses is that vessels can stop during a voyage for different reasons before arriving at their final destinations. Whenever a vessel stops moving or moves slowly, many AIS records are transmitted in clusters which cause noise and redundant data in the constructed trajectories.

The proposed solution includes using similarity between trajectories to predict traveling vessels' destination ports, therefore, in order to make the trajectories more comparable, a sampling step was added in the process of constructing the training data. There were two main approaches considered for trajectory resampling, namely sampling based on distance and time. When sampling based on a predefined distance, each subsequent point in a trajectory must be the same distance apart from the next. When sampling based on time, one position is extracted from a trajectory for every given unit of time. For instance, if sampling based on time with a six-hour sample rate, starting from the first point, every position within six-hour intervals are grouped and all positions within each group are dropped except for the first one. Both methods achieve the goal of making trajectories more comparable, however, sampling based on time simplifies, or reduces, the amount of data in each trajectory the most. It also provides an indication of trajectory duration implicitly through the trajectory length or the number of points in a trajectory.

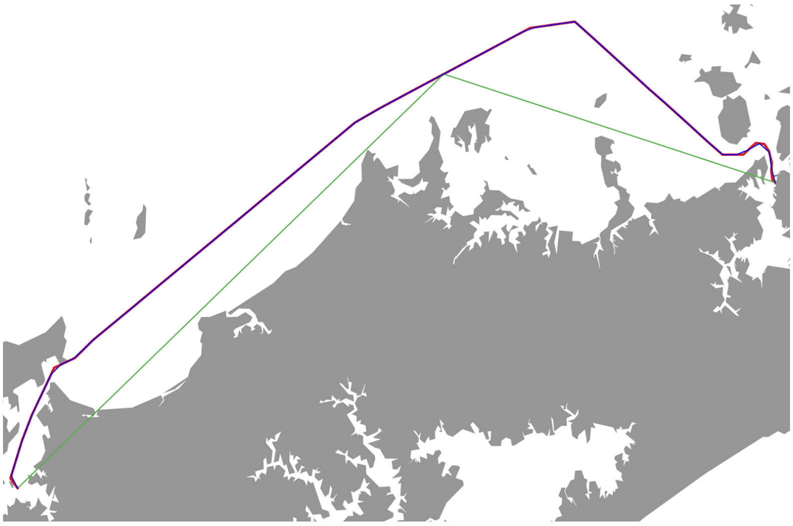


Figure 4.6: Example of a trajectory sampled by both distance (2 km) and time (6 hours). The red trajectory is not sampled, the blue is sampled based on 2 km in distance, and the green is sampled based on six hour time intervals.

For these reasons, throughout the rest of the implementation, sampling is done based on the time using a time interval of six hours. It is worth noting that for trajectories shorter than six hours, only the first and last point in the trajectory is returned reducing the trajectory to a straight line. In order to sample any trajectory based on either time or distance, a Golang package was written called “sampler” which can parse and handle both 3D trajectories including timestamps for time sampling and 2D trajectories for distance sampling. The complete code for this package can be found in Appendix B. Code listing 4.4 shows an extract from this package of the function used to resample trajectories based on time.

Code listing 4.4: Golang code from a sampler package written to sample a trajectory based on time.

```
// resampleTime resamples trajectory based on s.SampleRate given in hours.
// Extracts the first position within intervals based on sample rate
func (s *Instance) resampleTime() (string, error) {
        var err error
        trajectory, err := s.parse3DTrajectory()
        if err != nil {
                return "", err
        }

        intervals := s.getTimeIntervals(trajectory)
        reducedCoords := []geom.Coord{}
        coords := trajectory.Coords()

        // within each interval add the first coord to reducedCoords
        for _, interval := range intervals {
                var first *geom.Coord

                // get first coord in interval
                for i := range coords {
                        // roundTime uses s.SampleRate when rounding
                        coordInterval := s.roundTime(int64(coords[i][2]))
                        if coordInterval == interval {
                                first = &coords[i]
                                break
                        }
                }
                if first != nil {
                        reducedCoords = append(reducedCoords, *first)
                }
        }

        // if the last coord wasn’t the last in reduced, add it
        lastReduced := reducedCoords[len(reducedCoords)-1]
        if !lastReduced.Equal(geom.XYZ, coords[len(coords)-1]) {
                reducedCoords = append(reducedCoords, coords[len(coords)-1])
        }
        if len(reducedCoords) <= 1 {
                return "", errors.New("too few points in sampled trajectory")
        }
        reduced, err := geom.NewLineString(geom.XYZ).SetCoords(reducedCoords)
        if err != nil {
                return "", err
        }

        return geomwkt.Marshal(reduced)
}
```

The function listed in Code listing 4.4 is used in a batch process that samples every voyage’s trajectory and keeps the sampled voyages in a separate table called “sampled transition voyages”. The batch process extracts 5000 voyages at a time, samples their trajectories, and batch-inserts them into a separate table. By not mutating the original voyage data, different sampling methods can be applied and tested to find differences in trajectory comparisons. The final structure of the sampled transition voyages database table is described in Table 4.2

4.4.2 Most Similar Trajectory’s Destination (MSTD)

In order for the proposed solution to take into account both geographical trajectories as well as additional vessel information for predictions, in the training set, the trajectories have been abstracted into the categorical and numeric values Most Similar Trajectory’s Destination (MSTD), the similarity value for the MSTD, and then the length of the trajectory. The MSTD value is essentially an initial prediction of the vessel’s final destination purely based on geographical, or spatial, trajectories. This is similar to other approaches found in other studies such as Zhang et al. 2020. Purely spatial trajectory predictions works quite well when the trajectories are complete, or close to the vessels’ final destination, however, when a vessel has just recently departed a port for a long voyage, there are many possible destination ports and routes the vessel might take. The closer the vessel is to its final destination, the fewer possible candidate ports are there. When considering short trajectories of recently departed vessels, the most similar historical

Column	Type	Description
voyage_id	int	reference to the original voyage id
imo	int	identifier for the traveling vessel
mmsi	int	identifier for the traveling vessel
departure_port	string	UN/LOCODE of the vessel's departure port
departure_timestamp	int	Unix timestamp from when the vessel departed the departure port
arrival_port	string	UN/LOCODE of the vessel's arrival port
arrival_timestamp	int	Unix timestamp from when the vessel arrived at the arrival port
trajectory	geometry	3D linestring with longitude, latitude, and timestamp for each point

Table 4.2: Structure of the "sampled_transition_voyages" table.

trajectory is not likely to be a good estimation of where the vessel is traveling to. In these cases, relying on more general traveling patterns seem more appropriate. For example, the vessel's departure port, segment, and sub-segment are likely to be much better indicators as to the vessel's final destination. Therefore, the MSTD is only part of the final dataset used for predictions.

The MSTD for a given voyage is found by measuring the similarity between the given voyage's trajectory and every historical trajectory in the sampled transition voyages table. The most similar trajectory is found using a given trajectory similarity measurement, and its destination and similarity value are added to the final dataset. As described in Section 2.1.2, there are several different trajectory similarity measurements available, however, for this thesis the Symmetric Segment-Path Distance (SSPD) is used to calculate the MSTD values. However, the process and the dataset are structured in such a way that it is possible to use different similarity measurement methods in this process.

The MSTD value for a given voyage is calculated in the following steps:

1. Given a sampled voyage, fetch every historical voyage with the same departure port from vessels of the same segment and sub-segment.
2. Use a given similarity measurement method, SSPD in this case, to calculate the similarity between the given voyage's trajectory and every historical voyage's trajectory. The given similarity measurement method must return a similarity value. For SSPD, this value is a Haversine distance value.
3. Find the most similar historical trajectory or the trajectory with the smallest similarity value.
4. Extract the most similar historical trajectory's destination port as the MSTD value and extract the similarity value for future use.

Figure 4.7 shows an example of finding the most similar historical tra-

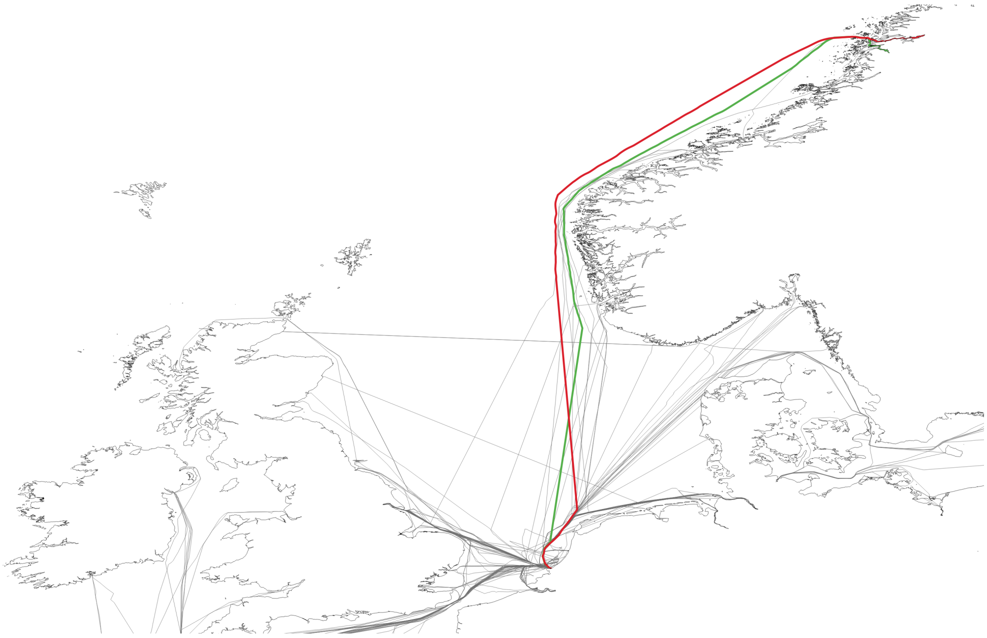


Figure 4.7: Example of MSTD for a given historical trajectory where the red line is the given trajectory and the green line is the most similar historical trajectory.

jectory (green line) for a given voyage (red line). The given voyage departed the port of Rotterdam and arrived in Mo i Rana, Norway. Every other historical voyage that departed Rotterdam from vessels of the same segment and sub-segment was then extracted and SSPD was used to find the most similar historical trajectory.

4.4.3 Building ML data training set

After the data has been collected, and the MSTD method has been implemented to translate geographical trajectories into categorical and numerical values, the last step is to collect data attributes from the initial data foundation and to calculate MSTD values for every historical trajectory. The final dataset is used to train a Machine Learning (ML) model that aims to predict the arrival port of voyages based on these values. Furthermore, to ensure that the training data is reflective of real-life scenarios, the full historical trajectories were divided into several incomplete voyages when calculating MSTD values. This ensures that the final model is able to predict the arrival port of traveling vessels before they reach their final destination. Thus, the process of constructing the final training dataset can be summarized in the following steps:

1. Extract the values listed in Table 4.2 from the sampled voyages.
2. Divide the trajectory up to four different smaller trajectories based on the length of the trajectory.
3. Calculate the MSTD for every trajectory.
4. Collect the values from each voyage as well as the MSTD, similarity value,

and length of the voyage trajectory for the training data.

The process of dividing a trajectory into multiple shorter lengths is relatively straightforward and is shown in Code listing 4.5. It is worth noting that trajectories containing less than four points are skipped as the constructed trajectories must have at least two points. Trajectories with exactly four points are divided into two parts instead of four for the same reason.

Code listing 4.5: Python code used to create incomplete voayges by dividing them into multiple lengths.

```
def get_incomplete_trajectories(df, parts=4):
    """divides trajectories in df into n parts

    Given a trajectory of length 8.
    We want to create the following sub-trajectories by incrementing with 8/4=2
     0  1
     0  1  2  3
     0  1  2  3  4  5
     0  1  2  3  4  5  6  7 # original
    """
    ret_df = pd.DataFrame(columns=df.columns)
    for _, r in df.iterrows():
        row = r.copy(deep=True)
        traj = row["trajectory"]
        length = len(traj)

        # we cant make trajectories of length 1
        # so we use half the number of parts
        if length == parts:
            parts = math.floor(parts/2)

        inc = math.floor(length/parts)
        if inc == 0:
            # skip trajectories shorter than parts
            continue

        for i in range(inc, length, inc):
            new_traj = traj[:i]
            if len(new_traj) < 2:
                continue

            row["trajectory"] = new_traj
            row["trajectory_length"] = len(new_traj)
            ret_df = ret_df.append(row, ignore_index=True)
    return ret_df
```

As an example, Figure 4.8 shows a voyage traveling from China to Argentina. The code listed in Code listing 4.5 was used to divide the voyage trajectory into four parts that are highlighted by the different colored segments of the trajectory.

The similarity value derived from the MSTD calculation and the length of the trajectory is also included in the training set so that a ML model can find correlating patterns between the length of the trajectory, the similarity value, and the MSTD value. The length of the trajectory indicates how long the vessel has

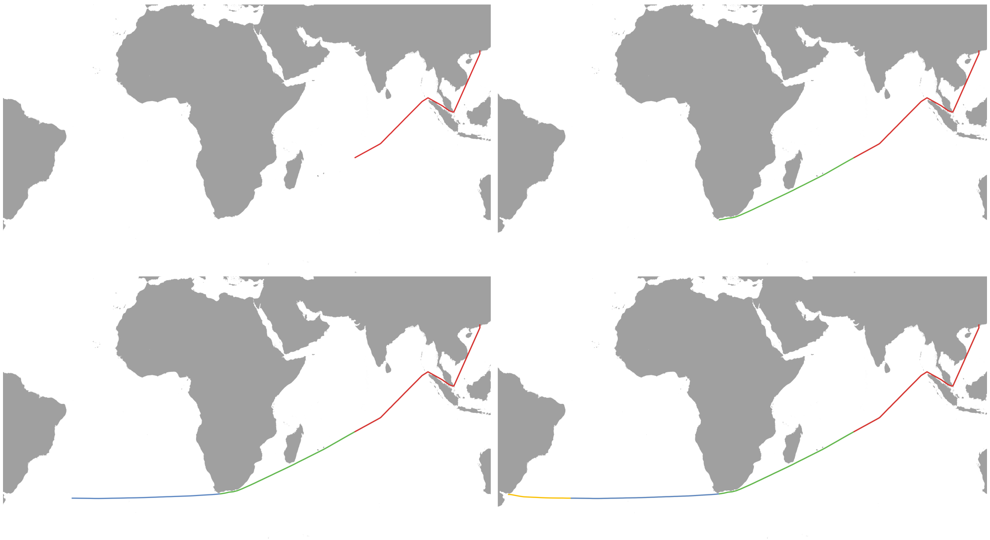


Figure 4.8: A voyage (ID 3305) from China to Argentina divided up into four subsets emulating incomplete voyages.

been at sea and thus how close it is to its destination. Therefore, both the trajectory length and the derived similarity value serve as weights for the MSTD value. The expected pattern from the constructed incomplete voyages is exemplified in Table 4.3. When the trajectories are short and the SSPD distance is high, it should indicate that the MSTD value is more likely to be wrong. When the vessel is closer to the destination, the trajectory length is longer and the MSTD is more likely to be correct. In the example of the vessel traveling from China to Argentina, the SSPD-based MSTD value was not correct until the last quarter of the voyage. The similarity value is also lower for the final entry which should indicate that the MSTD value for this row is likely to be correct which, in this case, it is.

Voyage ID	**SSPD-based MSTD**	**Arrival port**	**Trajectory length**	**SSPD dist.**
3305	INTUT	ARSLO	37	1520108
3305	MYPEN	ARSLO	74	150733
3305	BRRIG	ARSLO	111	454581
3305	ARSLO	ARSLO	148	148770

Table 4.3: Extract from ML training data exemplifying a voyage divided into four shorter voyages.

4.4.4 The final dataset – Summary

As described in Section 2.3, Maritime Optima AS (MO) provided the initial data foundation consisting of historical AIS data, a global set of shipping ports, vessel segmentation data, and transition events of vessels' navigational statuses. Historical AIS data was extracted, validated and MMSI numbers was mapped to the appropriate IMO number. The port data was extracted and filtered on a "visible" attribute which indicates that the port was deemed relevant by MO. Vessel transitions were extracted filtering only arrival and departure events, excluding events such as "detected" while the vessel segmentation data was purely copied without processing. Based on this initial data foundation, the final process of creating the dataset used in the analysis can be summarized in the following steps:

- Voyages are defined using time intervals provided by the vessels' AIS navigational status. They are constructed and stored in a voyage database table containing the full geographical trajectory, arrival and departure ports, and additional information for the traveling vessel.
    - The resulting table "transition_voyages" contains **1.7** million voyages.
- The voyage table's geographical trajectories are sampled, or simplified, based on a certain time interval to make trajectory comparisons easier.
- Every sampled historical trajectory is split into multiple parts to emulate incomplete voyages not yet arrived at a port. Furthermore, the MSTD is calculated for every one of these voyages. Trajectory similarity is defined using the SSPD algorithm, however, this data is interchangeable with other similarity measurements.
- Finally, the MSTD, similarity value, trajectory length, departure and arrival ports, and vessel segmentation information is collected and stored as the ML training data.
    - The resulting table "ml_training_data" contains **4.3** million voyages.

An overview of the process described in this chapter thus far is shown in Figure 4.9 from the data provided by Maritime Optima AS (MO) to the final ML training data. The final dataset is collected in the database table called ML training data, the attributes it contains are listed in Table 4.4.

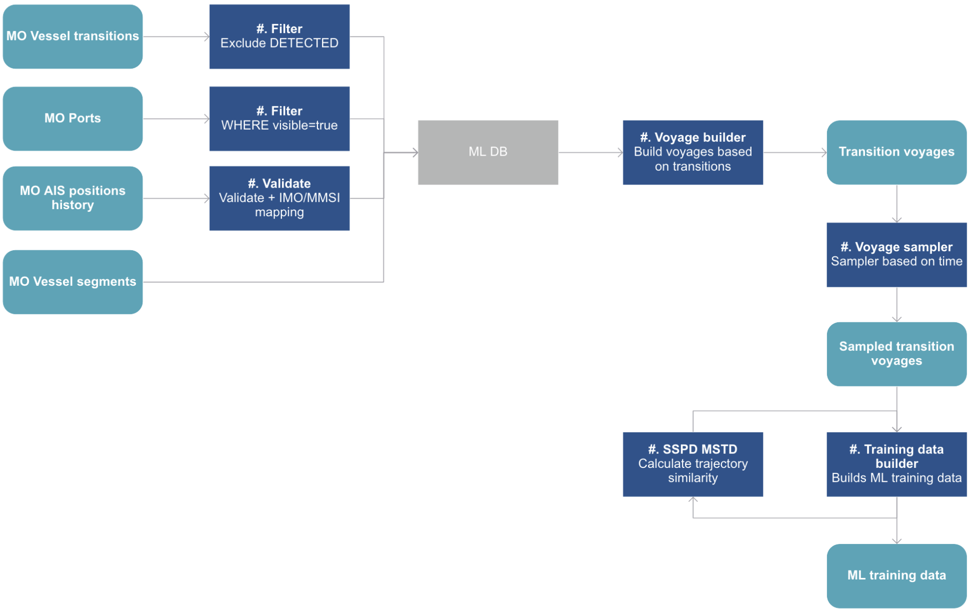

MO Vessel transitions
MO Ports
MO AIS positions history
MO Vessel segments
#. Filter Exclude DETECTED
#. Filter WHERE visible=true
#. Validate Validate + IMO/MMSI mapping
ML DB
#. Voyage builder Build voyages based on transitions
Transition voyages
#. Voyage sampler Sampler based on time
Sampled transition voyages
#. SSPD MSTD Calculate trajectory similarity
#. Training data builder Builds ML training data
ML training data

Figure 4.9: Overview of the process used to construct the dataset used in further analysis and ML

Column	Type	Description
id	serial int	unique identifier
voyage_id	int	the original voyage id from sampled transition voyages
imo	int	identifier for the traveling vessel
mmsi	int	identifier for the traveling vessel
segment	string	the vessel's segment
sub_segment	string	the vessel's sub-segment
departure_port	string	UN/LOCODE of the vessel's departure port
trajectory_length	int	number of points in the sampled trajectory
sspd_mstd	string	UN/LOCODE of the MSTD value for the voyage trajectory
sspd_dist	int	similarity value between the voyage trajectory and the most similar historical trajectory
arrival_port	string	UN/LOCODE of the vessel's arrival port

Table 4.4: Final structure of the ml_training_data database table.

4.5 ML-based training and destination prediction

After building the Machine Learning (ML) training dataset, the next part of the process included finding a ML model that suits the dataset, prepare and train it to predict values for the arrival port column in the training dataset. This section describes the process starting from the training dataset to the final trained prediction model.

4.5.1 Dataset imbalance

Due to the nature of vessel voyage patterns based on vessels of different types and sizes, there is a substantial imbalance in the number of voyages arriving at different ports. In other words, there is a severe imbalance in occurrences of arrival port values in the training set. Figure 4.10 shows the distribution of frequencies of voyages arriving at different ports, and as it shows, there is a significant imbalance in this distribution.

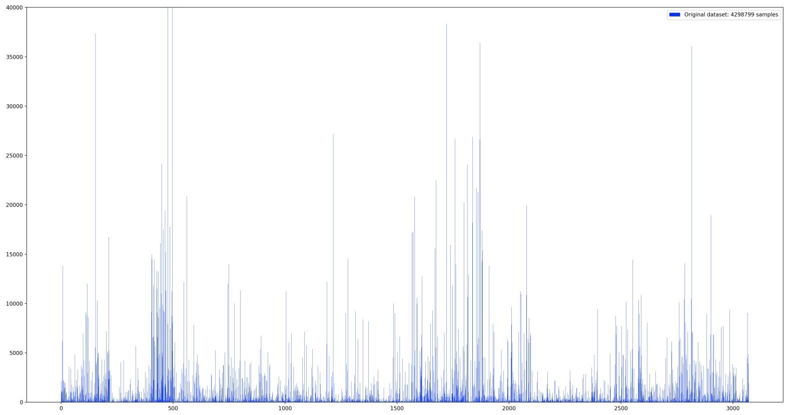


Figure 4.10: Graph showing the distribution of frequencies among the arrival port classes.

Because of the severe imbalance in the training set, simply applying oversampling techniques on the full dataset such as SMOTE (Chawla et al. 2002) to the original dataset increases the total data size to an unmanageable amount of **203** million samples. On the other hand, using a simple majority undersampling method removes most of the data, reducing it so almost no data is left. Figure 4.11 shows how different sampling methods perform using a limited version of the original dataset. As it shows, SMOTE (purple graph), oversamples the dataset massively, undersampling (red graph) reduces the dataset by *92%*, while SMOTE + ENN increases data size, but does a better job keeping relationships in the data.

Because of the significant outliers in the data, an ensemble of both under and oversampling was used to balance the dataset without generating a massive amount of synthetic data while also not removing too much information. The ensemble approach first undersamples classes that occur more than *20%* of the

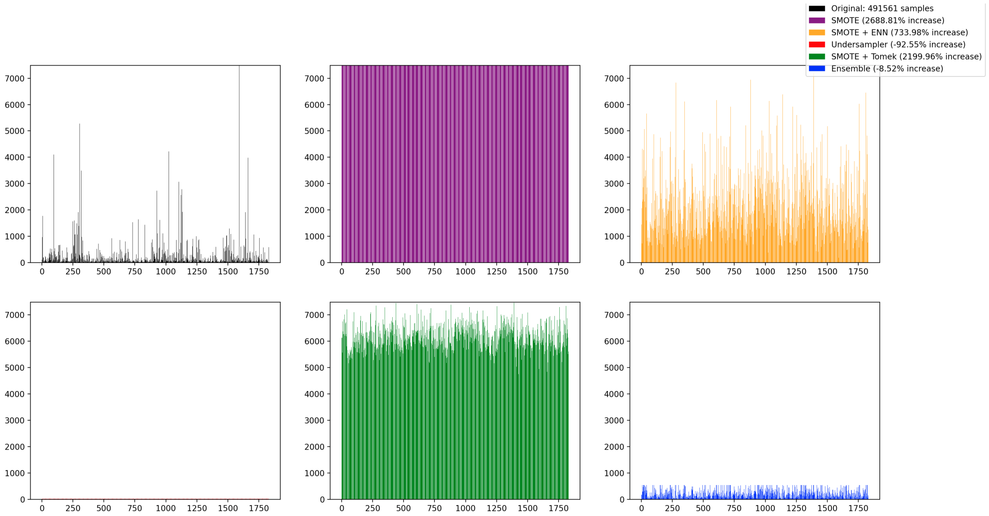


Figure 4.11: Different sampling methods compared based on the increase in data size.

most frequent class to remove the most severe spikes, applies *"SMOTE + ENN"* over and undersampling method to balance the dataset while still keeping important patterns in the dataset, then finally an undersampling step is added to flatten out frequency spikes to arrive at a final dataset size that is similar to the original. However, since SMOTE is used during this process, there are synthetic samples in the new dataset. These generated values are based on similar samples in the dataset, however, in the case of segment and sub-segment, specific sub-segments belong to certain segments, so when these are structured as separate values, SMOTE can synthetically generate invalid combinations of segments and sub-segments. To ensure the model doesn't waste time training on impossible segment and sub-segment combinations, the segment, and sub-segment values were combined into one segmentation value before encoding and balancing. This also reduces the complexity of the ML model structure as there is one fewer feature to consider. The segment and sub-segment values are concatenated using a delimiter, so they are easily divided again after training to further evaluate the results based on these values.

4.5.2 Categorical label encoding

In the training dataset, there are both numerical values as well as categorical values. Categorical values are values that are a subset of a finite number of possible values, while numerical values have infinite possible values. In the training data, the data concerning ports and vessel segments are examples of categorical values, and the length of the vessel trajectories and the similarity values derived from the MSTD value is numerical. The underlying problem of encoding categorical values and possible solutions have been described in Section 2.4.2, and as described, choosing an appropriate encoding method depends on the cardinality of the features in the dataset. For instance, there are more than 5000 possible ports in the data foundation, therefore, columns concerning ports such as arrival port, departure port, and MSTD have high cardinality. On the other hand, features such as segment and sub-segment have low cardinality, however, as these features have been combined into one "segmentation" feature, the combined feature has *107* possible values. *One-hot* encoding, therefore, seems impractical for either type of feature, thus, traditional label encoding was applied to the entire dataset before training. Code listing 4.6 shows an example of categorical values being label encoded as preparation before the training process.

Code listing 4.6: Example of training data before and after label encoding being applied.

```
[main] categorical training data features:
   departure_port  sspd_mstd  arrival_port  segmentation
0           CNRZH      CNWIT         CNSHG      dry_bulk-supramax
1           CNRZH      CNNDE         CNSHG      dry_bulk-supramax
2           CNRZH      CNSHG         CNSHG      dry_bulk-supramax
3           TRHER      TRZEY         RUNVS      dry_bulk-handysize
4           TRHER      TRZEY         RUNVS      dry_bulk-handysize

[main] encoded categorical training data features:
   departure_port  sspd_mstd  arrival_port  segmentation
0             574        593           464            27
1             574        557           464            27
2             574        574           464            27
3            3745       3666          2552            19
4            3745       3666          2552            19
```

4.5.3 Model selection

At this stage, the final data is fully processed and prepared for model training. However, first, the model that best fits the dataset must be chosen. In this process, several different classifiers were tested out with a smaller extract of the training data. Using Python libraries such as "Scikit-Learn" and XGBoost, the ML models tested are listed in Table 4.5.

Note that One vs. Rest (OVR) classifiers differs from the multi-class classification methods as they convert the multi-class problem to multiple binary classification problems. For instance, instead of predicting which port a voyage will arrive at, the OVR classifiers consider the perspective of a port so that the problem becomes: will the vessel arrive at this specific port? Moreover, the best performing model was the Random Forest (RF) model, followed closely by the two variants of XGBoost implementations. The RF and XGBoost models are somewhat similar methods. They are both ensemble decision tree methods meaning they configure multiple decision trees, or a forest, that vote on outcome values. This is based on the concept "wisdom of crowds" where multiple relatively uncorrelated trees acting as a committee is capable of outperforming a single decision tree classifier. In the RF method, the decision trees are constructed using bagging and boot-

Model	Description	Acc. %
Random Forest	Ensemble tree-based algorithm where trees are built using a random number of features.	91.6
XGBoost	Extreme gradient boosting. Ensemble tree-based algorithm where trees are constructed building on mistakes from previous trees..	89.9
One vs. Rest XGBoost	Train a binary XGBoost classifier for every possible arrival port.	89.5
k-Nearest Neighboor	kNN - samples look at their n neighbors to classify themselves.	77.4
Tensorflow + keras DNN	Deep neural network for multiclass classification. Nodes in sequential hidden layers are fitted based on an activation function.	50
One vs. Rest Multi-layered perceptron	Deep neural network -based binary classifier for every possible arrival port.	10

Table 4.5: Classifiers tested out in the model selection phase. Every classifier was trained using 50 thousand samples from the training data.

strapping as methods of randomizing the construction of decision trees either by randomizing what features to use when constructing a tree or what samples to use. The goal is that the trees gain their own unique perspective by being constructed in a particular way. In the XGBoost model, trees are created in a process called boosting wherein each boosting round a new tree is constructed based on the previous mistakes made by the previous trees.

Initially, the RF model was explored further for the final training process. However, the "RFClassifier" Python implementation seemed problematic for larger datasets as the memory requirements are considerable because of the size of the model as well as the high number of possible arrival ports which requires a large tree ensemble to learn. Furthermore, the Python implementation does not support out-of-core learning or incremental batch learning so it is less practical in the final training process. On the other hand, the XGBoost implementation seems more apt at handling larger data sizes as it uses less memory in general as well as it supports both out-of-core and incremental learning which provides more options in terms of computing resource requirements.

4.5.4 Configuration and parameter optimization

For different ML methods, there are usually many different configurations available to tune how the model learns. For the selected Extreme Gradient Boosting (XGBoost) model, based on the library's documentation, the most important parameters to configure are the following:

- *n_estimators* – the number of boosting rounds used. Default is 100.
- *max_depth* – the max depth per tree in the model. Default is 6.
- *subsample* – the fraction of samples from the training data considered when building each tree. Default is 1.0.
- *colsample_bytree* – the number of features to consider when building each tree. Default is 1.0.
- *min_child_weight* – the minimum weight required for a child node. Default is 1.0.
- *gamma* – the minimum loss reduction required to split a node in the tree. Default is 0.

The parameters *n_estimators*, *max_depth*, *min_child_weight*, and *gamma* helps control the complexity of the model, while the others introduce randomness to the tree building process. All of these parameters can be tuned to prevent overfitting and produce a well-performing model. In order to find the best parameters to use, the model was trained several times using different configurations on a subset of the original dataset in a process called hyper-parameter optimization. First, the parameters listed above were listed with a range of different values for each to test. Initially, these ranges were large to get an initial idea of what parameters fit the best. A random grid search was used to find a rough estimate of what values perform the best. In this process, 100 different configurations were randomly selected from all the possible permutations in order to get a feeling of what types of ranges fit the different parameters without attempting to train on all of them. The best result from this process provided insight into what general values seemed to work for the dataset. Code listing 4.7 shows an example of how parameters are defined and used in a random search. Next, a grid search was applied using smaller ranges based on the best result from the random search to further fine-tune the parameters. The grid search methods both use cross folder validation to arrive at the best performing configuration to consider overfitting in the final result, so the resulting parameters are a good place to start in the initial training process.

Code listing 4.7: Python example of parameters used for random grid search in hyper-parameter optimization process.

```python
random_grid = {
    "n_estimators": [100, 200, 300],
    "max_depth": [6, 8, 10],
    "subsample": [0.6, 0.8, 1.0],
    "learning_rate": [0.1, 0.2, 0.3],
    "colsample_bytree": [0.7, 0.8, 0.9, 1.0],
    "min_child_weight": [1, 2, 5, 8],
    "gamma": [0.0, 0.1, 0.2, 0.3],
}

# ...

best_params = best_random.random_search_cv(X_train, y_train, random_grid)
print("Best params from random search")
pprint(best_params)
```

```
# ...

def random_search_cv(self, X, y, param_grid, folds=3):
    # Random search of parameters, using X-fold cross validation,
    # search across 100 different combinations, and use all available cores
    self.classifier = RandomizedSearchCV(estimator=self.classifier,
                                         param_distributions=param_grid,
                                         n_iter=100, cv=folds, verbose=3,
                                         random_state=42, n_jobs=-1)

    # Fit the random search model
    self.classifier.fit(X, y)
    return self.classifier.best_params_
```

4.5.5 The training process

After finding appropriate parameters for the model, the next step was to conduct the training process. The process is straightforward, however, the size of the training set resulted in high computing requirements, especially in terms of available RAM on the running computer. XGBoost supports both iterative and external memory training routines, therefore, these alternatives were also evaluated to see what method works best. In total, the three alternatives available for the training process were: training the model iteratively by dividing the training data into multiple batches, using the computer's hard drive as memory using XGBoost's external memory mode, or simply training the model normally using a computer with sufficient hardware requirements. The training process was conducted using the full dataset containing **4.3** million voyages which were encoded and balanced beforehand. The training set was divided into *80%* training data and *20%* testing data which was used to estimate the performance of the model.

Moreover, the aforementioned approaches were initially tested using a computer with an *AMD Ryzen 7 2700* processor with 8 physical CPU cores and 8 additional virtual ones, an *NVIDIA GeForce GTX 1080* GPU, and 48GB of available RAM.

First, the full training set was used in the standard training process. However, this process demanded more memory than available on the machine. By trial and error, it was established that 48GB of memory was only sufficient to train on a subset of around 1 000 000 samples. Thus, the iterative training approach was tested using a limited subset of the available data in different batches. In this process, the full dataset was fetched, encoded, and balanced before it was split into batches. As the entire dataset was kept in memory during the batch process, the batch size was set to 600 000 for the iterative training process. For this process, the additional parameters *updater* and *process_type* were set to “refresh” and “update” respectively in order to ensure that the model correctly adapts to exposure to new samples. Code listing 4.8 shows the code used to prepare data into batches and iteratively train the model.

Code listing 4.8: Python functions used to batch train the XGBoost model.

```
def prepare_training_data(config):
    df = get_all_data()
    df = df.groupby(config["t_column"]).filter(lambda x: len(x) >= 20)

    X, y = encode(df, config)

    if config["sample"] == True:
        X, y = ensemble_sampler(X, y)
        inc = (len(X)-len(df))/len(df)*100
        print("[main] sampled data: {:.2f}% increase".format(inc))

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.20, random_state=42)
    batches = []
    for x in batch(range(0, len(X_train)), config["batch_size"]):
        batches.append(x)

    return batches, X_train, X_test, y_train, y_test

def batch_train_model(batches, X_train, X_test, y_train, y_test):
    params = {
        # ... other params
        "updater": "refresh",
        "process_type": "update",
        "num_class": len(np.unique(pd.concat([y_train, y_test]))),
    }

    # XGBoostClassifier is a custom wrapper around the XGBClassifier class.
    classifier = XGBoostClassifier(target_column="arrival_port", params=params)
    trained_model = None
    for t_indices in batches:
        # subset training data
        X = X_train.iloc[t_indices]
        y = y_train.iloc[t_indices]

        # eval set using the total training data
        dtrain = xgb.DMatrix(data=X_train, label=y_train)
        dtest = xgb.DMatrix(data=X_test, label=y_test)
        watchlist = [(dtest, ’eval’), (dtrain, ’train’)]

        # train the model
        trained_model = classifier.train_model(
            X, y, model=trained_model, num_boost_round=100, watchlist=watchlist)
        classifier.print_evaluate_summary(X_test, y_test, matrix=True)

    return classifier, X_train, X_test, y_train, y_test
```

Next, the external memory version of XGBoost was tested by running the training process in one batch using the machine’s hard drive as additional memory. In this process, the full dataset was fetched, sampled, and balanced before it was written to a *libsvm* file as recommended by the library’s documentation. The algorithm then produces a memory-optimized cache file on the computer’s disc which is then used in the training process. Code listing 4.9 shows how XGBoost’s “train” API can be modified to use external memory where the file name specifies “train.txt” as the input file using “dtrain.cache” as a temporary cache file.

Code listing 4.9: Python code showing how a XGBoost model can be trained using external memory.

```python
# internal memory version
# dtrain = xgb.DMatrix(data=X, label=y)

# external memory version
train_file = "train.txt#dtrain.cache"
dtrain = xgb.DMatrix("{:s}/{:s}".format(data_folder, train_file))

clf = xgb.train(self.params,
                dtrain,
                #... other parameters
            )
```

Finally, in order to establish the differences and to find the most optimal training process, a standard training process was tested on the full dataset on a more powerful computer. In this process, a virtual machine hosted on the collaborative company Maritime Optima AS (MO)'s cloud computing environment was used in the training process. This machine has 256GB of available memory and 32 virtual CPU cores. This was sufficient to train the full dataset and was the preferred method as it also allowed testing other ML models that do not support incremental or external memory training processes. Code listing 4.10 shows the code used to train the final model. An evaluation set was provided to the model to continuously evaluate it after each decision tree has been built in the model. It also allows for plotting performance metrics over each "boosting round" which can be helpful to detect when in the training process the model can become overfitted.

Code listing 4.10: Python code showing how the XGBoost model was trained in one iteration

```python
def train_xgb_classifier(df, target_column):
    params = {
        "use_label_encoder": False,
        "objective": "multi:softmax",
        "learning_rate": 0.1,
        "n_estimators": 100,
        "max_depth": 4,
        "subsample": 0.8,
        "gamma": 0.2,
        "eta": 0.2,
    }

    classifier = XGBoostClassifier(target_column=target_column, params=params)

    X_train, X_test, y_train, y_test = classifier.train_test_split(df, balance=True)

    # eval_set allows continuous evaluation during the training process
    eval_set = [(X_train, y_train), (X_test, y_test)]

    # fit the model (esr = early_stopping_rounds)
    classifier.fit_model(X_train, y_train, eval_set=eval_set,
                        esr=5, plot_results=True)

    return classifier, X_train, X_test, y_train, y_test
```

The results from the training processes are later discussed in Chapter 5.

Evaluation process

For all of the different training processes tested, the same evaluation process was used in order to establish the performance of the model. First, to detect if the trained model is overfitted, three-fold cross-validation applied during the training process. In this process, the model is trained three times using different parts of the training set as training and evaluation data. If the performance of each training process does not deviate significantly, the model is less likely to be overfitted and should be able to efficiently predict previously unseen samples. Moreover, plotting performance metrics for each iteration of the training process also provides insight into when the model stops learning and starts to overfit.

Based on ML conventions and related works, several performance metrics were used in addition to conventional accuracy. As already mentioned in Section 4.5.1, accuracy can be a misleading metric in some cases such as for imbalanced datasets. This is, however, a smaller issue after sampling has been used to balanced the dataset beforehand, but other performance metrics still provide more insight into the performance of the model. In addition to accuracy, the metrics, *logarithmic loss*, *classification error*, *precision*, *recall*, and *F1 score* were used. Furthermore, to gain more insight into the model performance on the particular dataset, accuracies per segment and sub-segment were also collected as listed in Code listing 4.11. Furthermore, this helps to gain insight into the traveling patterns of different vessel types as it pertains to the predictability of vessels of different types and sizes.

Code listing 4.11: Python code used to calculate accuracies per segment and sub-segment to gain insight into the predictability of different vessels.

```python
def column_accuracy(correct, incorrect, columns=["segment", "sub_segment"]):
    cr = correct.groupby(columns).size()
    cr.name = "correct"
    icr = incorrect.groupby(columns).size()
    icr.name = "incorrect"

    df = pd.concat([cr, icr], axis=1).fillna(0)
    df["total"] = df["correct"] + df["incorrect"]
    df = df.astype({ "correct": "int32", "incorrect": "int32", "total": "int32" })
    df["accuracy"] = df["correct"]/df["total"]

    # sort groups by accuracy if grouped by more than 1 column
    if len(columns) > 1:
        g = df["accuracy"].groupby(columns[0], group_keys=False)
        res = g.apply(lambda x: x.sort_values(ascending=False))
        return res

    return df.sort_values(by=["accuracy"], ascending=False)
```

4.6 Vessel destination prediction method summary

After the training process, the final trained model is saved and can then be used to predict the outcome of new samples of traveling voyages. To predict a traveling vessel's next destination, the steps described throughout this chapter must be replicated for that single vessel. Given the trained model, the overall prediction process for a single traveling vessel can be conceptualized using the following steps:

- The current trajectory of the traveling vessel is collected using AIS records ranging from the last transmitted *"MOORED"* status to its current position along with the id (UN/LOCODE) of the departure port where it was moored and the vessel's segmentation values.
- The vessel's trajectory is then sampled based on a predefined time interval, then compared to every historical outgoing trajectory from the same departure port from vessels of the same segment and sub-segment to establish the Most Similar Trajectory's Destination (MSTD).
- The vessel's segment, sub-segment, departure port, trajectory length, MSTD, and the MSTD similarity value is then passed to the trained XGBoost model that predicts the traveling vessel's arrival port.

Chapter 5

Results

In this chapter, the results from the proposed solution are described in detail. It describes different results from the different stages throughout the development process and presents the final results and metrics from the trained Machine Learning (ML) model. Furthermore, insights and interpretation into the results are gathered from experts in the shipping industry and described in order to determine the validity of the proposed solution.

5.1 Constructed dataset and ML problem formulation

The initial dataset was copied and validated from Maritime Optima AS (MO)'s AIS database. The database table *"vessel_positions_history"* was last updated in March 2021 and consists of **1.2** billion positional AIS records. Each vessel that transmitted positions belongs to a given segment and sub-segment that was made available by the *"vessel_segment"* table. This table contains **eight** different segment values, and **107** different combinations of segments and sub-segments. The provided *"ports"* data contains **5200** ports world-wide that all follows the UN/LOCODE naming standard. In total, as of March 2021, there were **6.4** million vessel transitions in the *"vessel_transitions"* table which was used to construct voyages.

This data formed the initial data foundation for the final processed Machine Learning (ML) training dataset. All the data that was copied and processed from MO's databases were processed in batches. Ports, segments, and transitions were quickly copied and processed, however, the **1.2** billion positional records took several days to migrate and validate. This was mostly because of the time required to validate coordinates and correctly map MMSI and IMO values. Throughout this process, the latest identifiers and timestamps were fetched from the dedicated project database to only update data that occurred after the latest records already processed. In this way, this process was idempotent so that running the process multiple times did not affect existing data. This made the system simple to update throughout the development process and as many records as possible were used in the final approach only limited by the thesis time limitation.

5.1.1 Voyage definition and construction

Based on the initial **6.4** million vessel transitions, **1.7** million voyages were initially constructed by finding positional records transmitted from a vessel between subsequent departure and arrival transitions. The resulting voyages were, therefore, defined based on transitioning AIS statuses that indicate the vessel is moored or moving. As a consequence of this definition, the quality of the resulting trajectories is very much affected by how well the AIVDM/AIVDO protocol is followed by the traveling vessels. Since the navigational status attribute is manually inputted by the vessel's captain or crew, the resulting trajectories are prone to human error but result in more complete voyages disregarding intermediate stops for purposes such as bunkering.

As an example, Figure 5.1 shows a voyage from China to Argentina where the vessel stopped at Singapore, most likely to bunker. In the chosen voyage definition, the beginning and end of the voyage are defined based on input from the vessel's captain which results in a voyage starting from China and ending in Argentina. Further implications and consequences of the chosen definition are later discussed in Chapter 6.

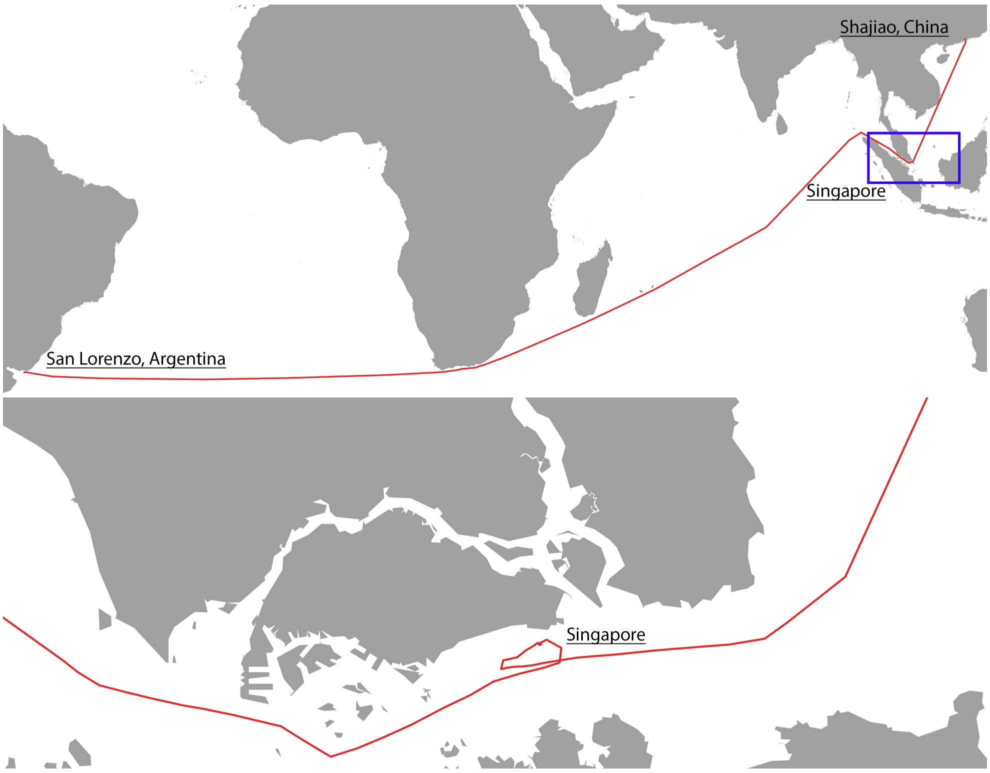


Figure 5.1: Transition voyage from China to Argentina that visits the port of Singapore exemplifying the properties of the chosen voyage definition.

The **1.7** million voyages constructed using the vessel transitions were sampled based on 6-hour intervals and collected in “sampled_transition_voyages” that formed the foundation for trajectory similarity measurements. In the process of constructing the final dataset, these sampled voyages were divided into multiple incomplete voyages up to a factor of four. The resulting training dataset collected in the table “ml_training_data” consisted of **4.3** million voyages.

5.1.2 Trajectory similarity and MSTD

Using the foundation of the sampled trajectories, each trajectory was compared to every other trajectory departing the same port to calculate the Most Similar Trajectory’s Destination (MSTD). The MSTD value was used primarily as a method of abstracting geographical trajectories into categorical and numerical values that a Machine Learning (ML) model could work with. This process converted a voyage’s geographical trajectory into MSTD, the similarity value to the most similar trajectory, and trajectory length. Thus, the MSTD value served as an initial prediction purely based on geographical trajectory similarity measurements using Symmetric Segment-Path Distance (SSPD). The SSPD method was chosen for its ability to effectively handle different lengths and shapes of trajectories when estimating similarity. Furthermore, the approach proposed by Zhang et al. 2020 the SSPD method performed the best out of the algorithmic approaches evaluated, although, their own Random Forest (RF) based approach performed the best. However, the way the training data is structured, the trajectory similarity method of choice is completely interchangeable with others. The only requirement for a given trajectory similarity measurement is that it also produces a similarity value that serves as a weight for the MSTD value.

MSTD as an initial prediction seemed to be a decent initial indicator as to where the vessel would be arriving. In total, there were **4 306 271** entries in the final training data generated where exactly **1 423 476** of which has the same arrival port and MSTD value. Thus, it can be assumed that the purely spatial prediction using incomplete sampled historical voyages based on SSPD was *33%* accurate. In other words, when using an algorithmic prediction approach based on purely spatial trajectory similarity measurements, voyage destinations can be predicted correctly one-third of the time. This formed a baseline accuracy to beat with the ML-based solution.

5.1.3 ML data preparation

After the final training dataset was built, it was discovered that in terms of arrival port frequencies, the dataset was imbalanced thus making it harder for ML models to learn. Although some models can better handle dataset imbalance, a sampling approach was used to balance the dataset before training to support different ML models. Several different sampling approaches were evaluated, however, the traditional over and undersampling methods either produced massive amounts of synthetic data or removed almost all the original data which was shown in Figure 4.11 in Chapter 4. Thus, an ensemble sampling method of majority undersampling and "SMOTE+ENN" was employed to balance the dataset before training. Figure 5.2 shows the results from the ensemble sampling method that uses a combination of under and oversampling techniques. As Figure 5.2 shows, using a subset of the full dataset, the final result is 8% smaller than the original dataset, is a lot more balanced, but still has differences in class frequencies that persisted from the original dataset.

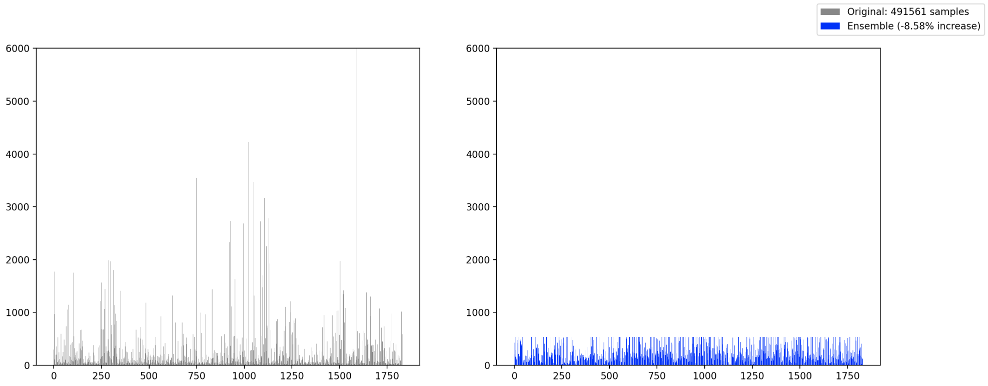


Figure 5.2: Final ensemble sampling method (right) compared to original dataset (left) where the final ensemble produces a dataset similar in size to that of the original.

5.2 Model training and prediction performance

After the training dataset was constructed, encoded, and balanced, the model was trained using different approaches as described in Section 4.5.5. This section describes the results from the training processes, the final approach used, and the resulting model's performance and predictions resulting from the evaluation process described in Section 4.5.5.

5.2.1 Training process

As described in Section 4.5.5, multiple training processes were evaluated in order to find the most appropriate method of training a larger model on an extensive dataset. For the Extreme Gradient Boosting (XGBoost) model, three different training processes were evaluated in this process.

First, the iterative approach was evaluated by training the model in batches of *600 000* samples at the time. This approach seemed to work as intended, however, it was discovered that during subsequent training batches, the performance of the model dropped off for each iteration. It seemed as if the model did not handle continuous training of the same model as well as it does when training one model from scratch using the complete dataset. Furthermore, the parameter *"early_stopping_rounds"* was used in the other approaches as a method of telling the model to stop training if it does not see any improvements after the given number of rounds. When this parameter is set using the iterative approach, the model can stop producing new trees before it has constructed the total number of trees allowed by the *"n_estimators"* parameter. Since the first iteration can produce a model with fewer trees than allowed, the next iteration fails as the number of allowed trees does not match the previous model's actual number of trees. Although there are ways around this issue, as using the early stopping rounds parameter is useful to avoid overfitting, the iterative approach did not seem the most appropriate during the development process.

Next, it was attempted to train the model using the external memory, or the "out-of-core" memory version of XGBoost. In this approach, the XGBoost library is provided a *libsvm* file which it converts to an optimized matrix format that is kept on the computer's file system. However, all attempts at training the model using external memory were unsuccessful as the training process consumed all of the running computer's available memory and resulted in a "bad allocation" memory error. There seems to either be a misconfiguration or an underlying issue with the Python library used in the implementation. However, since the expected results from this approach should be the same as training the model in one iteration on a capable computer, these issues were not further looked into. However, it could be beneficial to use this option to reduce the resource requirements for the training process for future use. Therefore, it could warrant more investigation for future work.

Finally, the entire dataset was used to train the final model in one iteration on a computer capable of running the process. The training process ran over the course of two days and consistently required around 200GB of memory. The vast memory consumption could be somewhat reduced by not evaluating the model during the training process which is appropriate for future training processes after the model has been trained and the training configuration has been validated. As described in Section 4.5.5, an extra copy of the training and test datasets was kept in memory to continuously evaluate and monitor the training process.

5.2.2 Performance

During the training process, the performance of the model was continuously evaluated to measure logarithmic loss and multi-class classification error. Figure 5.3 shows these metrics plotted over each boosting round in the training process. Both graphs start converging at 100 decision trees have been constructed at around **1.5** log loss and around **0.3** classification error. This corresponds to around **70%** accuracy. Since the graphs have not completely converged, it is possible to either increase the learning rate parameter or increase the number of estimators in the tree, although it seems as if the graphs are very close to converging, so it might not increase performance noticeably and increases the risk of overfitting. As there is very little difference between the performance on the training set and evaluation set, it indicates that the model is not overfitting, however, it might indicate that the model is over-optimistic. This could occur when there are several similar samples in the training and the test datasets and could be a consequence of the sampling techniques used to balance the dataset.

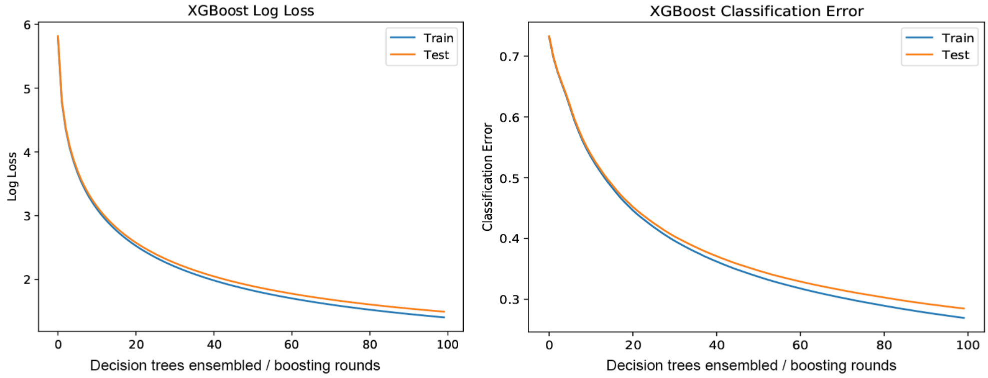


Figure 5.3: Logarithmic loss and classification error metrics tracked per boosting round in the training process.

After the training process finished, the test dataset was used to make predictions to further evaluate the results. From the resulting predictions, accuracy was calculated to be **72%**, and a class report was generated that shows more metrics for each possible class, or encoded arrival port, that might provide more insight into the model's performance than accuracy. Code listing 5.1 shows a summarized output from this class report showing the metrics precision, recall, f1-score, and support for each class as well as the aggregated mean values from all of the classes. F1-score is based on precision and recall and is particularly appropriate for measuring performance on imbalanced datasets. As Code listing 5.1 shows, the f1-score does not deviate much from the estimated accuracy of **72%**, or **0.72**. This indicates that the accuracy value is reliable and is not biased by dataset imbalance.

Code listing 5.1: Class report based on prediction results from the test dataset. The performance of the classifier is evaluated per class by using precision, recall, f1-score, and support.

```
[XGBoostClassifier] Class Report:
             precision    recall  f1-score    support      pred
0             0.378049  0.240310  0.293839      258.0     164.0
1             0.816850  0.810909  0.813869      275.0     273.0
2             0.722222  0.541667  0.619048      312.0     234.0
3             0.672727  0.377551  0.483660      294.0     165.0
...                ...       ...       ...        ...       ...
3067          0.824675  0.849498  0.836903      299.0     308.0
3068          0.833922  0.778878  0.805461      303.0     283.0
3069          0.773050  0.762238  0.767606      286.0     282.0
3070          0.614035  0.557325  0.584307      314.0     285.0
```

```
...           ...       ...       ...         ...       ...
avg / total   0.718698  0.715150  0.712737  878049.0  878049.0
```

Lastly, in order to ensure the model is not overfitted, a three-fold cross-validation process was employed. Code listing 5.2 shows the results from the three folds that the model was trained on which used a weighted F1-score as the performance metric. It is recommended, or common to use more folds ranging from five to 10, however, because of the long training time and time limitations, only three folds were used. As shown in Code listing 5.2, the mean F1-score across each fold was **73%** which is slightly higher than the initial training round. Lastly, as described in Section 2.4.4, since the standard deviation (noted as "std. dev." in Code listing 5.2) is very low (9.025×10^{-5}), the model is likely to not be overfitted.

Code listing 5.2: Output from 3-fold cross validation.

```
Folds:      [0.73393717, 0.73398, 0.73414641]
Mean:       0.7340211926682213
Std. dev.:  9.025433684824043e-05
```

5.3 Prediction results

After the model was trained and evaluated, *20%* of the total training dataset was used to evaluate the model. This evaluation process resulted in around *880 000* example predictions. These predictions were further analyzed to discuss the impact and meaning of the different features used in the dataset. These results are presented in this section.

5.3.1 Feature importances

An added benefit of using a tree-based model such as the Extreme Gradient Boosting (XGBoost) or Random Forest (RF) model is that they can provide insight into the importance of features, or attributes. In a decision tree-based ensemble, when constructing a tree, the training data is analyzed to find the best features to make splits, or branches, in the trees. After the training process, the models can then produce a ranking over what features best divided the dataset best. This is referred to as feature importance.

Table 5.1 shows an overview of the produced feature importances after the XGBoost training process. As it shows, the most important feature was the MSTD value at a ranking of 0.44 out of 1.0, followed by the vessel's departure port, segmentation value, and then the similarity value and voyage length. This analysis can further help decide if features are worth dropping from the dataset, and insight into what attributes are good indicators during voyage predictions. As mentioned in Section 4.5.1, the attributes "segment", and "sub-segment" were combined into one segmentation value in order to ensure that no invalid segment

Feature	Importance
sspd_mstd	0.443659
departure_port	0.226288
segmentation	0.180907
sspd_dist	0.083816
trajectory_length	0.065331

Table 5.1: Feature importances based on the XGBoost decision tree ensemble process

and sub-segment combinations could be generated by sampling methods. A disadvantage of this is that the feature importances of the two attributes are lost in favor of the combined value. However, from test runs made during the development process with and without sampling, the importance of segment and sub-segment were usually ranked where segmentation is in Table 5.1 with sub-segment being more important than segment.

Furthermore, the results from test dataset predictions were analyzed to find the impact of the attributes that mostly served as weights for the MSTD value, namely, the similarity value (*sspd_dist*) and trajectory length. Code listing 5.3 shows an output from the evaluation process which shows that the distance value was smaller, on average, for correct predictions while trajectory length did not considerably differ between correct and incorrect predictions. It makes sense that the distance, or similarity, value is lower for correct predictions as the more similar the most similar historical trajectory is, the more valuable the MSTD value is. For instance, if a voyage's most similar historical trajectory has a *sspd_dist* of 0, it is following an exact path of a previous voyage. In this case, the similarity value for correct predictions was on average around *43%* lower than for incorrect predictions. For the trajectory length, it would make sense that the longer the voyage had traveled, the easier it would be to predict its destination, thus, the length should be longer for correct predictions. However, this is not the case for these predictions. This could be explained by the fact that shorter voyages might be easier to predict than long voyages for small vessels. For instance, it is presumable that, passenger vessels with very short but frequent trajectories are very easy to predict, thus reducing the average length for correct predictions. To confirm this hypothesis, further investigations into the specific segments and sub-segments are required.

Code listing 5.3: Mean values of similarity value and trajectory length for correct and incorrect predictions.

```
mean ssp_dist for correct predictions:          115642.48170757179
mean trajectory_length for correct predictions:  17.662729492637958
mean ssp_dist for erroneus predictions:          201713.174255885
mean trajectory_length for erroneus predictions:  18.841843522761508
```

5.3.2 Segment predictability

As it relates to research question 2 (Section 1.5), the *880 000* predictions from the test dataset were further analyzed in search of patterns in the predictability of different types of vessels. These results also serve to gain further insight into the value of the performance metrics. Figure 5.4 shows a bar chart of the initial accuracy of predictions per segment, and it shows that there are some differences in accuracy per segment overall, but most of the segments have a similar level of predictability. For example, vessels of the segment “other” were the easiest to predict and had the highest accuracy of *76%*. This is likely to be caused by different types of passenger’s vessels that lie within this segment. These vessels produce many predictable voyages as they travel between a few ports with a high frequency. Furthermore, the “other” segment also includes very specialized vessels that are limited in terms of possible destination ports.

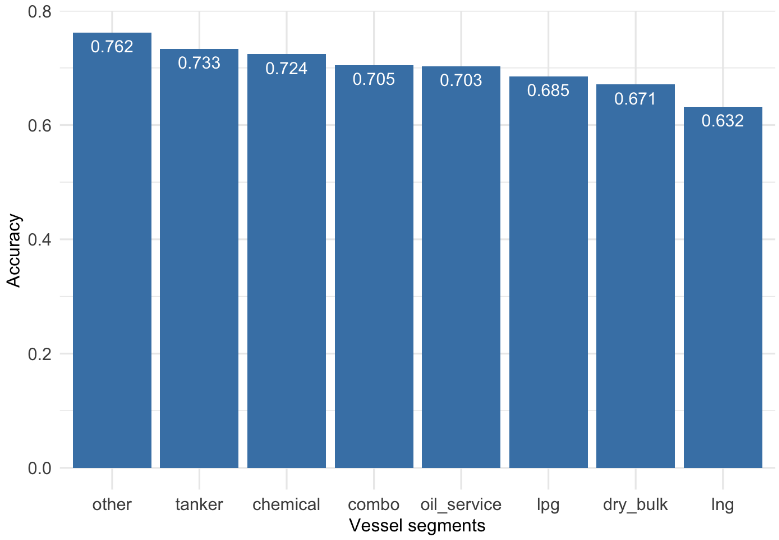


Figure 5.4: Accuracy of predictions from test set per segment.

As Figure 5.5 shows, and as expected, the accuracy of the passenger-related sub-segments was very high. Since these are so high in frequency and have shorter trajectories, they may be the main cause that the average trajectory length was lower for correct predictions than incorrect ones. On the other hand, container and car “roll-on/roll-off” (RORO) vessels travel longer distances less frequently but were also relatively predictable.

Another segment that could affect the average trajectory length and similarity values for correct predictions is the oil service segment. The oil service vessels should be easy to predict as these vessels travel to oil platforms and often back to the same or another nearby port. However, for these vessels, their trajectories would have been harder to consider as they often do not use the “moored” AIS navigational status when arriving at oil platforms. This can lead to very long trajectories that are hard to compare to others, therefore, these vessels should rely more on the departure port rather than the MSTD related values.

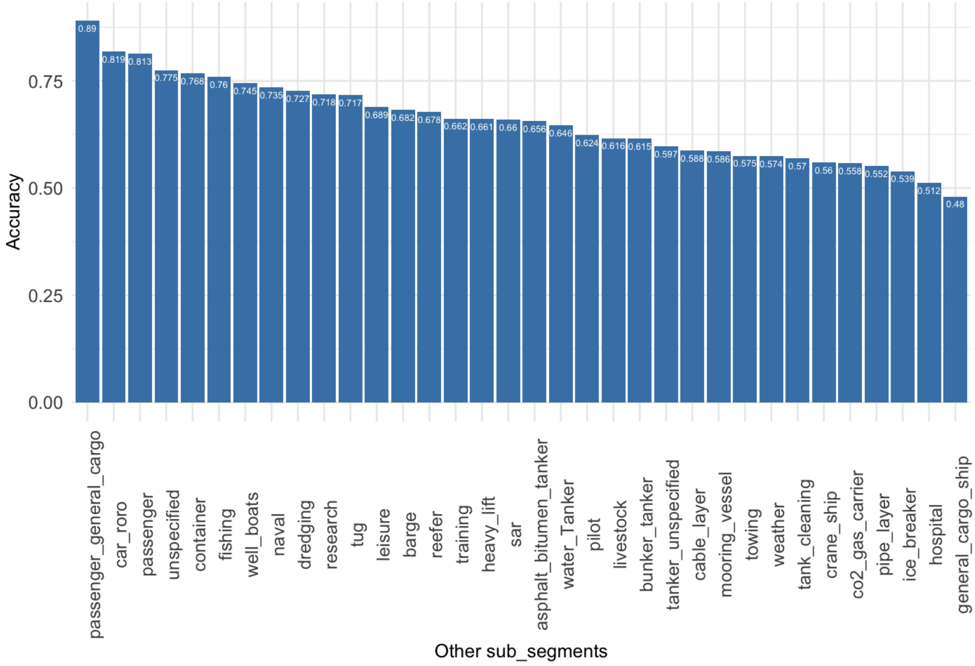


Figure 5.5: Accuracy of predictions per sub-segment within the “other” segment.

The dry bulk cargo industry is one of the primary segments focused on by MO, and Figure 5.6 shows the accuracy per sub-segment for the dry bulk cargo segment. The dry bulk sub-segments are based on the vessels’ cargo capacities and sizes, however, as Figure 5.6 shows, there seems to be little correlation between vessel size and accuracy. The two most accurately predicted sub-segments are large vessels, however, they are followed closely by the smaller sub-segments, and the two least predictable types are some of the largest. Thus, the uniqueness of the sub-segment value itself had more impact on predictions than the implied size and capacity of the vessels.

The prediction results for tanker sub-segments show similar results as to the dry bulk ones, however, some other segments do seem to show that size and capacity indeed might be correlated to predictability in different ways. For instance, in the chemical segment, the two largest sub-segments have the highest accuracies of *90%* and *85%*, however, the remaining sub-segments do not show much difference correlated to size. There seem to be a slight correlation in chemical vessels that show that larger vessels are easier to predict than smaller ones, however, for other segments the opposite correlation seems to occur. The Liquefied

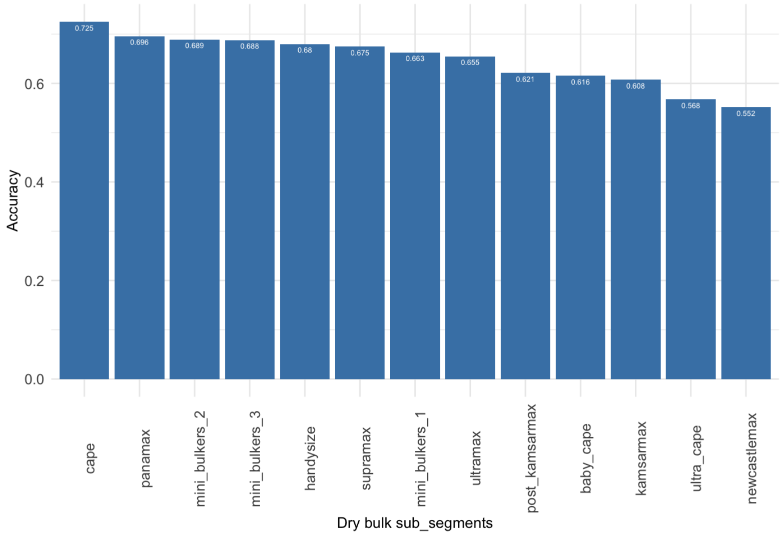


Figure 5.6: Accuracy of predictions per sub-segment within the “dry_bulk” segment.

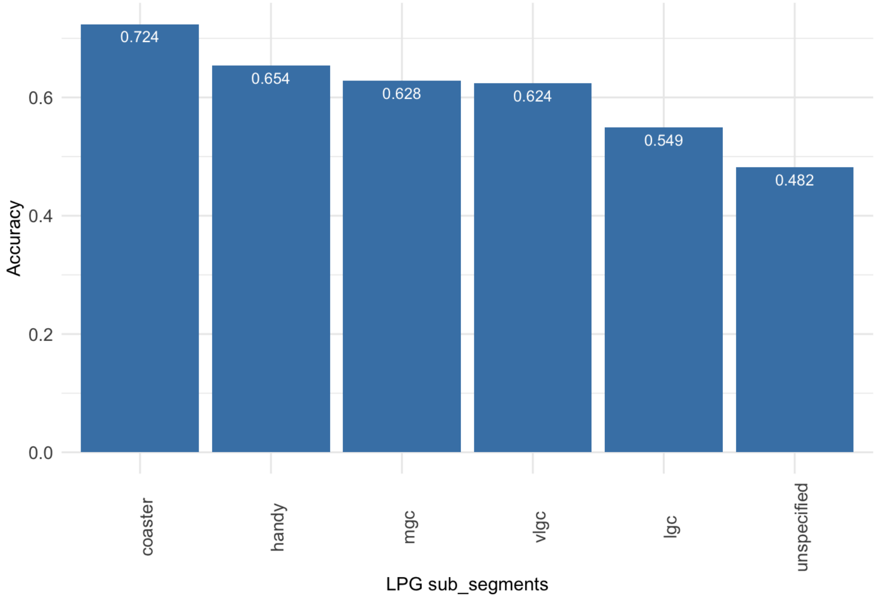


Figure 5.7: Accuracy of predictions per sub-segment within the “LPG” segment.

Natural Gas (LNG) and Liquefied Petroleum Gas (LPG) vessels have the highest correlation between size and accuracy, but in the opposite direction compared to the chemical vessels. Figure 5.7 shows that the three smallest LPG sub-segments *coaster*, *handy*, and *MGC* have the highest accuracy, while the two largest sub-segments *VLGC* and *LGC* have lower accuracies. This is similar to that of the LNG vessels (Figure 5.8) where the largest sub-segments *QMax*, and *QFlex* are harder to predict than the smaller sub-segments. This is quite unexpected as these vessels are very limited in possible loading and discharging port. However, it could be explained by there being very few samples of these vessels in the dataset compared to the smaller vessels.

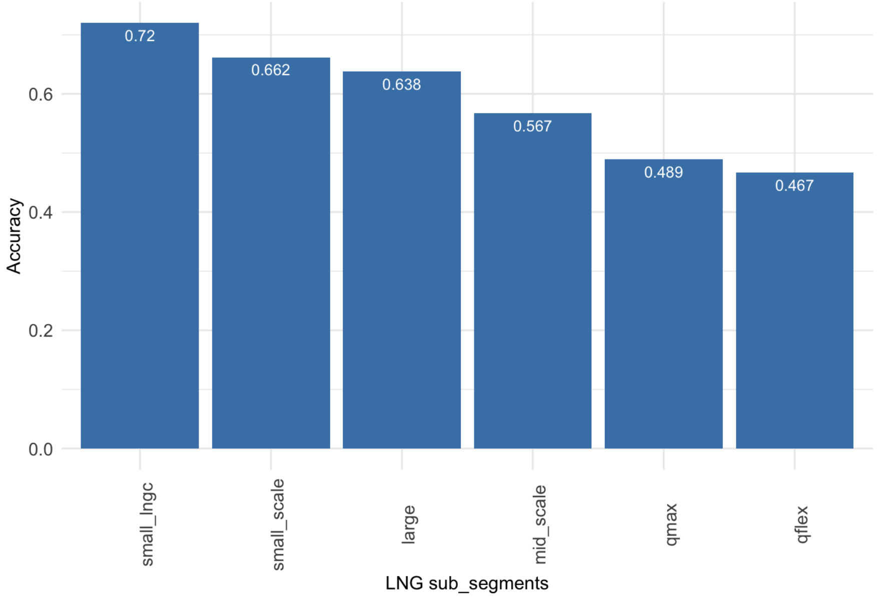


Figure 5.8: Accuracy of predictions per sub-segment within the “LNG” segment.

Another interesting segment to analyze is the combo segment. These combination vessels can serve multiple functions in that they can carry different types of cargoes. In Figure 5.4, the combo segment showed a mid-range general accuracy level, however, when looking into the sub-segments, there are substantial differences in accuracies across the different types of combo vessels (Figure 5.9). The “Klaveness Combination Carriers” (CABU) and “Oil-Bulk-Ore” (OBO) vessels have the highest accuracies. However, there are only 12 CABU vessels and 5 OBO vessels in the world, or in Maritime Optima AS (MO)’s vessel database. On the other hand, there are 4700 chemical product tankers in the world that were also quite predictable. These vessels drive the general accuracy of the combo vessels up in Figure 5.4 as the remaining sub-segments have substantially lower accuracies. It does, however, make sense that combo vessels are generally difficult to predict as they serve multiple functions which results in them having more possible destination ports they can load and unload at.

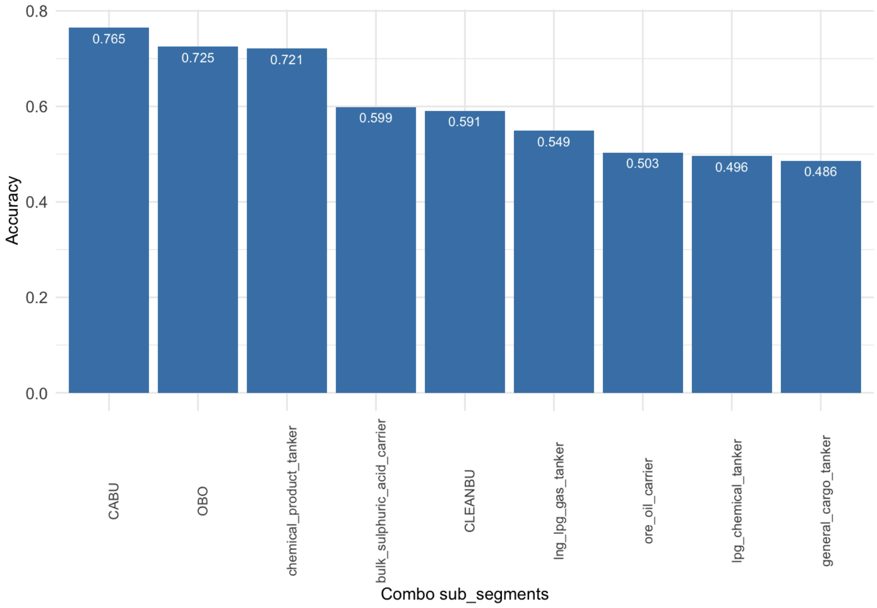


Figure 5.9: Accuracy of predictions per sub-segment within the "combo" segment.

In regards to `RQ 2` (Section 1.5), vessel segments and sub-segments seem to have a substantial impact on the predictability of vessels. As shown in Table 5.1, the vessel segmentation had a feature importance close to that of the vessel's departure port. Furthermore, as discussed throughout this section, there are differences in accuracies for different segments and sub-segments, therefore, the vessel segmentation, with sub-segments in particular, had a significant impact on the predictions in the test dataset used during the evaluation process.

In regards to `RQ 2a` the most predictable segment overall was the "other" segment (Figure 5.4). This was not entirely surprising as the sub-segments including passenger's vessels are very predictable (Figure 5.5). Moreover, the tanker, chemical, and combo vessels were similar in their accuracy levels, while LPG, dry bulk, and LNG vessels were slightly less predictable. The sub-segment "chemical product tanker" drove the accuracy of the combo segment up to a similar level to that of the tanker and chemical vessels. This can be explained by the fact that this specific sub-segment overlaps into the two other segments. In other words, several tanker and chemical vessels are also present in the "chemical product tanker" combo sub-segment, so the accuracies are expected to be similar between the specific sub-segment and the tanker and chemical segments.

In response to `RQ 2b`, and as mentioned earlier in this section, there seems to be some correlation between vessel size, capacity, and predictability, however, this only seems to be the case for some segments while for others, the uniqueness of the sub-segment value was the more important factor than the implied size or capacity. Thus, in regards to `RQ2b`, the prediction results do not conclusively indicate that larger vessels are more predictable than others. This is likely to be caused by there being few voyages available for larger vessels traveling further as the original dataset only contains one and a half years of historical data.

5.4 Applications and validity

After the final training process, the resulting model is capable of predicting the future destination ports of traveling vessels with a general accuracy of *72%*. This section describes the intended usage and applications of the developed model as well as validation from experts in the industry.

5.4.1 Usability

As summarized in Section 4.6, the process of predicting a single vessel's future destination port consists of four steps. First, the current traveling trajectory is constructed by fetching the positional AIS records from the last detected "moored" navigational status was transmitted to the last transmitted position. Next, this trajectory must then be simplified as described in Section 4.4.1. Then the Most Similar Trajectory's Destination (MSTD) of the must be calculated using the SSPD method with the traveling trajectory and every other historical trajectory departing the same port. The vessel's MSTD, segmentation, the distance returned from the SSPD method, and the length of the trajectory can be used to predict the vessel's next destination. The final trained model is saved to a file so it can quickly be loaded when making predictions. Thus, a program can be written that reads the trained model, receives an outgoing voyage, and predicts its next destination port.

In regards to re-training the model with new data, two approaches can be used. The simplest but more time-consuming approach is to completely retrain the model after a substantial amount of new data is available. The training process takes around two days to complete using the complete historical dataset on a capable computer. Another approach could be to use Extreme Gradient Boosting (XGBoost)'s support for iterative, or continuous learning as described in Section 4.5.5. After the training process has completed, the XGBoost model can be saved to file for future evaluation and predictions.

Finally, since the proposed solution can predict a vessel's future destination ports, it could also be applied to forecast the availability of vessels. By providing the model with the current trajectories of all traveling vessels in the world, the model's output would indicate how many vessels of different segments will be positioned at different ports around the world. Given a method for estimating the time taken for each vessel to reach their predicted destination port, the model can be used to indicate what vessels will be positioned at a port in a given time interval. The model itself has no aspect of time, or ETA, however, there are existing methods and tools available today that can estimate how long it takes for vessels to travel between ports such as established distance tables[1] or software provided

[1] https://sea-distances.org/

routing estimators. For instance, Maritime Optima AS (MO) offers a routing estimator to their customers that is capable of finding the shortest path between any given two positions across the seas. The proposed solution could therefore be integrated with this tool to forecast the supply of vessels at different ports in a given time interval.

5.4.2 Expert validation

Furthermore, in order to establish the validity of the proposed solution from a commercial perspective, a select number of shipping experts were interviewed in order to establish the validity of the process taken and the final results. These experts were contacted via the collaborative company Maritime Optima AS (MO) which has a substantial network in the shipping industry. They were presented with the proposed solution as well as the steps taken throughout the development process and asked questions in a semi-structured manner in order to gain insight into their perspective on the following topics:

- Existing methods used to obtain predictions of vessel positions or vessel availability.
- Aspects of the thesis' solution that may prove to be valuable, and areas to improve before commercial consideration.
- Validity of the proposed voyage definition and possible alternative approaches.
- The impact of vessel segmentation and possibly other information that could provide more insight into vessels' voyage patterns thus improving predictions.

The people interviewed hold executive positions in well reputable companies and are very experienced shipping professionals. In respect to their privacy, their names, positions, or related companies will not be disclosed in this thesis, only a summary of the obtained information is presented.

Existing methods of obtaining information

One interviewee explained they did not use many digital tools in their decision-making processes. They had been using Maritime Optima AS (MO) to gain some overview, but mostly relied on non-digital methods of obtaining information. It was clear the source considered the most reliable was information and analysis provided by shipping brokers. These brokers provide information and predictions regarding the relevant segment's market, cargo, and vessel supply. It was also clear that information provided by any digital solution would require extensive testing and validation before it could be considered as any form of replacement or addition to the information provided by trusted brokers. It was also suggested that tools such as proposed in this thesis would probably have high value for the brokers themselves, to aid in their information gathering processes.

Another interviewee explained they were extremely reliant on making market predictions for multiple vessel segments. In addition to input from shipping brokers, they have spent considerable effort toward conducting their own analysis using several sources of information including historical AIS. Therefore, they rely more on internally conducted analysis than external ones and expressed high interest in similar analysis to that of the one presented in this thesis.

Valuable aspects of the proposed solution, and areas of improvement

From the imagined usability described in Section 5.4.1, the aspect of obtaining a forecast of vessel supply in different ports and regions seemed to be the most promising aspect of the thesis from a commercial perspective. The interviewees explained that the aspect of cargo supply is, to some degree, quite predictable as the production of different supplies is quite cyclic and there are many detectable factors that indicate ebbs and flows in productions. Since the areas of high cargo supply are known, knowing how many competing vessels are available in these areas could be valuable information because it helps operators decide whether to focus on certain cargoes in the different areas or not. Interviewees with different commercial motivations all expressed interest in the proposed solution as a competitor analysis tool as well as general input into different commercial analyses.

Some interviewees expressed some skepticism of a generally applicable prediction model as it was thought that an immense amount of data would be required to make accurate predictions. They further expressed the need for extensive testing and validation before it would be considered valuable to them. One interviewee especially expressed that it was of no interest to them to study segments outside of their own vessels, and suggested that a model specially trained for their segment could potentially be of higher value to them.

Other interviewees also expressed the complexity and broadness of making reliable vessel supply predictions as well as vessel destination predictions. However, they were adamant that any input of information is of value in the complete picture. For example, if a vessel's arrival port prediction is wrong it is still useful input as it might allude to the intended destination region or country. Therefore, as long as the performance and limitations of a prediction model are well known, its input is valuable even if it is not extremely accurate.

Validity of the proposed voyage definition

The chosen vessel voyage definition was explained and the example described in Section 5.1.1 was presented to the interviewees for their evaluation. Based on their response, it seemed that the suggested voyage definition would be a technically correct solution as it is based on the vessels themselves expressing via AIS that they are moored at different ports. However, another promising approach was suggested by one of the interviewees and corroborated by another. They suggested that the navigational status could be ignored if additional port information were to be used. For instance, when vessels stop at loading ports, they are likely to load or depart, and at known unloading ports the visiting vessels are likely to unload, or arrive. This could be implemented by using additional port information in combination with the clustering approach described in Section 4.3.1. An imagined issue with this approach, however, was that some bunkering ports also serve other functions such as unloading, therefore, it could be more difficult to separate bunkering visits from unloading activities. This approach would attempt to deduce the context surrounding a vessel's port visit, and although it seems promising, it requires additional port information, and potentially more analysis into the vessels' trajectories during port visits.

Additional vessel, or voyage information for prediction improvement

Lastly, the interviewees were asked what vessel or voyage features they imagined could gain insight into voyage patterns and subsequently improving destination predictions. In addition to the vessels' segment and sub-segment, they thought that the loading condition of the vessels would have a substantial impact on predictions. For example, if a vessel is loaded, it has fewer possible destination ports as it must arrive at a discharge port to unload. On the other hand, if the vessel is unloaded, or in ballast condition, it will probably visit a loading port next. It is possible to estimate a vessel's loading condition by looking at the vessel's current draft in static AIS messages. The draft of a vessel describes how deep the vessel is traveling, in meters, in the water. This value is higher when the vessel carries cargo, and lower if it is in ballast. Thus, if the information regarding vessels' loaded conditions were known during a voyage, the model could easily be trained to recognize these patterns and most likely be more accurate.

Moreover, it was suggested that ports' depth restrictions could be considered when making predictions. As the current draft or depth of the vessel is known, it can only arrive at ports that are deep enough to receive it. This type of information could also serve as valuable input to determine what ports are relevant shipping ports, as mentioned in Section 2.3.3. In terms of predictions, larger vessels have a fewer number of ports that have the capability of receiving them in contrast to smaller vessels, thus, it could limit the number of possible arrival ports for some vessels. Lastly, based on the experts' opinions, other factors such as the current season, or month, could also have an impact on predictions as voyages are quite cyclic in nature, especially for some cargoes such as grain which is harvested at certain times of the year at different locations in the world.

Chapter 6

Discussion

In this chapter, a summary of the thesis is provided, followed by discussions of the proposed solution, the field of study, possible applications, and the approach's validity both in terms of academic and commercial value. Finally, the limitations of the thesis and proposed future work are presented and discussed.

6.1 Summary

This thesis has investigated the topic of destination prediction for vessels in the shipping industry by using historical Automatic Identification Systems (AIS) data and additional vessel information. AIS is a globally adopted tracking system that transmits all commercial vessels' geographical and navigational information similar to that of GPS.

The thesis objective was to develop a methodology for predicting traveling vessels' future destination ports on a global scale unrestricted by specific vessel types, geographical regions, or time intervals using historical AIS data. The thesis was based on a collaboration with a technological maritime start-up company Maritime Optima AS (MO) who provided the data foundation used throughout the thesis, as well as access to experts to validate the solution. In doing so, the thesis sought to answer two primary research questions:

1. How can AIS data combined with specific vessel details be applied to predict future destinations of maritime vessels?
2. What is the impact of vessel segmentation on prediction methods, or vessels' general predictability?

Related work within the area of vessel destination or trajectory prediction was investigated to determine to what extent existing literature had already answered the research questions. It was found that the majority of related work was motivated by collision avoidance for safety reasons, anomaly detection to detect vessels deviating from established shipping lanes, automated collision avoidance systems to be installed on autonomous vessels, or short-term trajectory predictions to aid in port management and scheduling. Existing works motivated by these factors did not consider the future destination port of the traveling vessels, but rather the vessels' specific positions or future trajectories in a short time interval ranging from minutes to a few hours.

The few related studies that considered future destinations of vessels were almost all limited, or exclusively tested on a specific region or area such as the Mediterranean sea. One study was found that considered a general port destination prediction approach, however, this study, presented in Zhang et al. 2020, exclusively considered the geographical information provided by the AIS standard, and did not consider additional vessel information such as the type, size, or capacity of vessels. Since the research questions were not fully answered by the existing literature, the thesis goal was refined to developing a global and general vessel destination prediction method that is capable of considering more than purely spatial voyage information such as vessel segments and sub-segments as provided by the collaborative company Maritime Optima AS (MO).

In order to use spatial voyage trajectories derived from AIS data in the final prediction method, the thesis formulated a voyage definition that determined what conditions had to be true to consider a vessel arrived at a specific port. Based on related works, a clustering-based approach was initially evaluated that detected "clusters" of AIS records close to shipping ports using the Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. When positional records were transmitted in a high enough density close to a port, the vessel was considered arrived. However, this approach was later abandoned in favor of a voyage definition that considered the vessels themselves expressing an arrived state via the navigational status "moored" in the AIS data. The latter approach was favored as it ignored intermediate port visits during a voyage such as when vessels stopping to refuel at bunker ports, therefore, it produced more realistic voyage characterizations of higher predictive value, taking into account where the vessel finally unloads cargo and considers itself moored.

Using the predefined voyage definition, historical AIS data ranging from December 2019 to March 2021 provided by MO was constructed into **1.7** million voyages and trajectories defined as positional records transmitted between subsequent departures and arrivals. These voyages formed the initial training data to be used to train a Machine Learning (ML) model to predict voyages' arrival ports. In order to consider specific vessel information, voyage information, and spatial trajectories, a method of structuring spatial trajectories as categorical and numerical values was proposed.

In this approach, every historical trajectory was compared with every other trajectory outgoing from the same departure port from vessels of the same type in order to find the most similar historical trajectory. The Symmetric Segment-Path Distance (SSPD) algorithm was used to determine trajectory similarity. The trajectories had been simplified prior to this comparison by only using one point at every six-hour interval in each trajectory in order to make comparisons easier.

Furthermore, in the process of trajectory comparisons, each voyage was divided into at most four incomplete versions of the same trajectory to emulate realistic voyages not yet reached their final destination. The most similar historical trajectory's destination port (MSTD), the value indicating the similarity value of the trajectories, and the trajectory length became the categorical and numerical values representing a voyage's spatial trajectory. The final training set only contained categorical and numerical values, consisted of **4.3** million incomplete voyages, and formed the final training dataset used to train a ML model.

An Extreme Gradient Boosting (XGBoost) ML model was then configured and trained to predict a voyage's arrival port by considering the vessel's segmentation, or type, departure port, MSTD, MSTD similarity, and trajectory length, or duration. The training process used *80%* and *20%* as training and evaluation data respectively and achieved an accuracy score of **72%**, and an F1-score of **0.734** validated by additional metrics and cross-folder validation. The **20%** of the dataset used to evaluate the initial training process resulted in around *880 000* example predictions that were then analyzed in order to gain insight into *research question 2* (Section 1.5). It was found that the segmentation value had a large impact on the model's performance, and some vessel segments were easier to predict than others. For some segments, accuracies for underlying sub-segments defined by vessels' size and capacities seemed to indicate some correlation between vessel size, capacity, and predictability, however, in other segments, this did not seem to be the case.

Finally, in order to determine the validity of the proposed solution, several high-ranking commercial shipping actors were interviewed. They provided a valuable perspective into the commercial validity of the thesis and had numerous suggestions for possible applications and future improvements.

6.2 Research questions

This thesis has aimed to answer two main research questions, as listed in Section 6.1, as well as a number of sub-questions as specified in Section 1.5. This section describes how each research question was answered as part of the proposed solution or preliminary literature review.

6.2.1 RQ 1: How can AIS data combined with specific vessel details be applied to predict future destinations of maritime vessels?

The existing literature was unable to fully answer this research question which further motivated the developed model proposed in this thesis. Thus, the thesis proposes a method of predicting the future destinations of vessels based on historical AIS and specific vessel details. Vessel voyages were defined and trajectories were constructed using historical AIS records. These trajectories were structured as categorical and numerical values by making initial predictions purely based on the spatial trajectories by calculating the Most Similar Trajectory's Destination (MSTD). The resulting training dataset was extended to include additional vessel details such as the vessels' segments and sub-segments. Thus, any classificationoriented ML model could be trained to predict voyages' arrival ports.

RQ 1a: What prediction methods can be used to predict vessel destinations?

In addition to the thesis' proposed method, existing literature showed a few methods capable of predicting destination ports. The only study found unlimited by specific geographical regions developed a Random Forest (RF) -based trajectory similarity measurement method that was used to find a traveling vessel's most similar historical trajectory's destination port similar to that of the MSTD value used in this thesis. They also used the frequencies of port visits to normalize the predictions. In the solution proposed in this thesis, their ML-based trajectory similarity method could replace the SSPD method when calculating the MSTD value in the training dataset and when making predictions.

In terms of ML models, when the problem of destination prediction is formulated as a classification problem it seems that the most viable models are tree-based ensemble methods such as the Random Forest (RF) or Extreme Gradient Boosting (XGBoost) models. In contrast, for short-term trajectory predictions, many different models were applied in related works. For these predictions, nearest neighbor search-based approaches were common as well as a variety of feedforward neural networks.

RQ 1b: What information can be used to predict vessel destinations?

The related work showed that purely spatial attributes in historical AIS data had been used to make predictions regarding vessels' future trajectory or destination. A few studies used the vessels' heading and speed as well as their geographical coordinates when making predictions, however, it was most common to only consider trajectories derived from geographical coordinates when making predictions. In the thesis' proposed solution, the vessels' departure ports and vessel segmentation proved to be highly impactful on destination predictions.

Furthermore, shipping experts interviewed explained that in addition to vessel segments and sub-segments, vessels' current drafts (depth underwater) in addition to port restrictions can indicate where vessels will travel. Large vessels are particularly affected as there are few ports that are capable of receiving and loading them. Weather also has a large impact on vessels' traveling patterns but usually does not affect vessels' final destination port as this is already decided before the voyage begins. The experts also explained that seasonality may be an impact factor as some wares are only exported during particular seasons. However, in areas affected by ice, it may also impact vessels' voyage trajectories as some areas are unnavigable for most vessels during winter months such as the Northeast passage.

RQ 1c: To what extent do methods proposed in existing work vary in scope of applicability?

Time extent Based on the results from a review of the current literature, most prediction methods did not consider future destination ports, but rather vessels' future short-term trajectories. The development of such methods was often motivated by security improvement and used to detect possible collision scenarios. As collision scenario detection is only relevant in shorter time-frames, these prediction methods were limited to restricted time intervals ranging from minutes to a few hours at most.

Geographical extent As most related studies are motivated by goals such as security improvement, port management, and anomaly detection, they are not only limited by temporal extent but also geographical extent. For collision detection, the geographical area is not very relevant because the developed methods should be applicable to any given area, port, or region. Moreover, anomaly detection studies were usually limited to specific geographical regions in order to reduce the amount of noise or irrelevant data. For example, for detecting illegal, or irregular, fishing activity, only fishing vessels in a particular area are considered. The few longer-term prediction methods discovered that did consider logistics were also mostly limited by a single geographical region. Seemingly, this was often a result of limited access to global historical AIS data, or the studies themselves were conducted in collaboration with a specific maritime organization. Only one study was found to consider destination predictions on a global scale independent of both geographical and time limitations.

Data depth In terms of the broadness of data considered for related studies, most studies only considered geographical data. Some studies considering collision detection additionally took advantage of additional navigational attributes of the AIS data such as the vessel's heading, Course Over Ground (COG), and Speed Over Ground (SOG). The few studies that considered destination port predictions were dependent on port data, however, they only considered vessels' spatial trajectories in their predictions and ignored specific vessel details such as their types or segments.

The fact that most related work is generally motivated by safety improvement and collision detection reflects the original intent behind the AIS initiative. The main intention behind the AIS initiative was not economical, or commercial, in nature, but rather implemented for safety and navigation reasons. However, in recent years, the commercial shipping industry has begun using AIS for commercial purposes as it has become a trusted source of information. Thus, it is probable that more studies will focus on AIS for destination prediction and logistics in the future.

RQ 1d: How can the validity of predictions made based on different prediction methods be established?

There were many different validation approaches taken in the existing literature. The most common validation method included using some manner of *k-fold cross-validation* with multiple performance metrics such as F1-score, Mean Distance Error (MDE), and accuracy. Error measurements based on distances were mostly used in short-term predictions that required high positional accuracy, however, it was also applied for a few papers that considered destination prediction as well. In these studies, the distance from the predicted destination port was measured from the actual destination port. This provides further insight into the level of error for incorrect predictions. For example, if a predicted arrival port was wrong but very close to the actual arrival port, the trajectory-based prediction did still perform quite well. This is a good candidate for future work for this thesis as the proposed evaluation process did not provide much insight into the level of error for incorrect predictions.

6.2.2 RQ 2: What is the impact of vessel segmentation by type, size, or capacity on prediction methods, or vessels' general predictability?

During the model evaluation stage in this thesis, almost *900 000* sample predictions were produced and analyzed in order to determine accuracy levels across vessels of different segments and sub-segments. Moreover, as a tree-based ensemble model was used, a sense of feature importance was produced during the training process. The resulting feature importance showed that the combined segmentation value had feature importance of *0.18* or *18%* which was close to that of the departure port value at *0.23*, or *23%*. A full overview of feature importances is shown in Table 5.1.

These feature importances indicate that the vessel segmentation values play a significant part in the predictability of vessels. This is expected since many shipping ports are specialized and different countries and regions produce, export, and import different types of goods and services. The prediction model proposed in this thesis was generally applicable to any vessel of any segment. However, it could be possible that separate models trained on different vessel segments could provide more insight into the predictability of different vessel segments. For large vessels of specialized segments such as LNG vessels, there are very few possible loading and discharging ports. Therefore, a ML model could perhaps be more effective when trained on a segment-specific, low cardinality dataset with only a few possible arrival ports.

In the generalized approach presented in this thesis, there are also more samples available for smaller vessels than larger ones as there are fewer larger vessels in the world. Larger vessels usually travel over longer distances making their voyages less frequent. Therefore, the model could have had a harder time learning patterns for larger vessels, thus, being somewhat biased toward smaller vessels even though larger vessels would have been assumed to be more predictable than smaller ones.

RQ 2a: What types of vessels are more predictable than others?

Based on the resulting predictions from the evaluation process, the accuracy levels of the eight different segments are shown in Figure 5.4. The segment “other” had the highest total accuracy of *76%* which resulted from a high number of very predictable passenger vessels as well as container and car carriers that were also quite predictable. The tanker and chemical vessels were slightly more predictable than dry bulk, Liquefied Petroleum Gas (LPG), and Liquefied Natural Gas (LNG) vessels.

Combination vessels within the segment “combo” also showed a high accuracy since it includes a sub-segment that overlaps into the tanker and chemical segments otherwise, the other sub-segments showed low prediction accuracy. The oil service vessels also showed a high accuracy level caused by a singular sub-segment including oil platform supply vessels. It is logical that combo vessels are hard to predict as they can serve multiple functions giving them a broader range of possible loading and unloading ports.

RQ 2b: Do larger vessels travel in more predictable patterns than smaller vessels?

The resulting predictions from the evaluation processes showed that there seemed to be some correlation between size, capacity, and predictability, but only for some segments. This was investigated by looking at specific sub-segment accuracies for sub-segments that were based on the size or capacity of vessels. The sub-segments within the chemical segment somewhat indicate that larger vessels were easier to predict, however, LNG, and LPG vessels indicate a stronger correlation in the opposite direction where the smaller sub-segments showed higher accuracies. For segments such as tanker and dry bulk, no strong correlation was found between size, capacity, and predictability. Thus, based on the proposed general prediction method, there is no strong indication of larger vessels being easier to predict than others.

There are likely to be multiple factors responsible for this conclusion. Firstly, there are fewer voyages available for larger vessels in the *1.5* years of AIS data available than for smaller vessels because longer voyages take longer to complete. Thus, the models know fewer samples of large vessels in the training dataset and might be slightly biased toward smaller vessels.

Moreover, the longer trajectories are more likely to contain intermediate port visits along a voyage as vessels stop to refuel at bunker ports. Such trajectories would then be harder to compare with other voyage trajectories that did not refuel at the same bunker port. This is less likely to impact smaller vessels traveling short distances.

Lastly, there is a trade-off between the size of the vessels and whether trajectory predictions or port frequency is the best indicator of where it is traveling to. For smaller vessels on shorter voyages, trajectory predictions are more reliable than port frequencies as the trajectories are shorter and quite comparable, but there are many different possible arrival ports that the vessel can arrive at. For larger vessels, trajectory predictions are harder because of the length of the voyages, but there are few possible arrival ports that can receive large vessels. As Table 5.1 shows, the model found MSTD to be the most important feature. This could further indicate that the model could be somewhat biased toward smaller vessels which are likely to have better MSTD estimations because of more comparable trajectories. The fact that trajectory-based predictions are more effective for smaller vessels is further alluded to by the trajectory length being shorter for correct predictions than for incorrect predictions.

6.3 Limitations and application challenges

This section aims to disclose and discuss important application, or implementation, challenges as well as other possible impact factors and limitations that might have affected the analysis conducted throughout the thesis.

6.3.1 Vessel voyage definition

As summarized in Section 6.1, the thesis composed a specific voyage definition that was used to construct voyages from historical AIS data. This definition was an important aspect of the proposed solution as it forms the foundation of what voyages are and how resulting voyage predictions are characterized. Related work granted little insight into this area as few studies considered individual voyage predictions, however, one approach was proposed in Zhang et al. 2020 that involved using the DBSCAN algorithm to detect clusters of positional AIS data transmitted by individual vessels close to ports. A similar approach was investigated in this thesis (Section 4.3.1) where clusters were detected, mapped to their closest port, and labeled as an arrival at the port. The main disadvantage of this approach was that it defined vessels bunkering (refueling) as arrivals as it cannot distinguish between vessels stopping at ports to load or unload and vessels stopping close to ports because they are held up or bunkering and vessels stopping to load or unload cargo.

An alternative approach was proposed in this thesis where the navigational status attribute in the AIS data was used to determine when a vessel has arrived at a port. A vessel is considered to arrive when the status is set to “moored” close to a port. This navigational status is a manual input attribute that the captain or crew onboard a ship manages. This means that when the signal is set to “moored”, it is the captain of the vessel that considers the vessel as arrived at a port. Thus, the alternative definition trusts the vessels themselves to manage their moored and moving statuses which have the advantage of producing more commercially viable voyages but are affected by human error and lacking standardization. This latter approach was the chosen voyage definition throughout the thesis as the cluster-based solution proved difficult and time-consuming to configure in order to get a good voyage definition for all vessels, while the latter definition, on average, produced high-quality voyages.

From expert validation, some additional opinions were given on the topic of voyage definitions. One promising suggestion was given that mostly revolved around using additional port information to determine what the purpose of port visits was. For instance, vessels are likely to load at loading ports, likely to unload at unloading ports, and likely to bunker at bunkering ports. Thus, a third alternative could have been constructed using a combination of the clustering approach with additional port data to determine why vessels stop at different ports. Although the required information per port was not available when developing the thesis' proposed solution, it shows promise as a future improvement on the thesis work.

6.3.2 Geographical trajectory abstraction and MSTD

Another challenge discussed throughout the thesis is the method of which to consider both geographical trajectory and additional vessel and voyage information in a Machine Learning (ML)-based prediction method.

In this thesis, a vessel's spatial trajectory is reconstructed in the categorical value Most Similar Trajectory's Destination (MSTD) and the numerical values MSTD similarity, and trajectory length. The MSTD value is a preliminary guess of the vessel's destination purely based on its trajectory by comparing it to every historical trajectory outgoing from the same departure port. The MSTD is found using a trajectory similarity measurement algorithm called Symmetric Segment-Path Distance (SSPD). This method is especially apt at handling trajectories of different lengths and shapes which was beneficial for comparing incomplete voyages not yet arrived to complete historical trajectories. In the training dataset which consisted of **4.3** million incomplete voyages, the MSTD value corresponded exactly to the actual arrival port for *33%* of the voyages. This means that a purely spatial algorithmic approach could be *33%* accurate using this approach.

In regards to related work, the method proposed by Zhang et al. 2020 was a purely spatial trajectory similarity-based approach. Their Random Forest (RF) based ML approach achieved an accuracy of *67%*. Although the accuracies are not comparable as two different voyage definitions were used, there could be a method of combining both approaches to construct a more efficient geographical trajectory abstraction and ultimately improve the final prediction method. This combined approach could also be improved by more data attributes such as vessel segmentation and the loading condition of vessels which could result in a highly accurate MSTD value.

6.3.3 Dataset imbalance

During the preparation stage for ML training, it was discovered that the dataset suffered from a significant imbalance in terms of the frequency of arrival port classes. When ML models are trained on imbalanced datasets, the models see more examples of some outcomes than others which can lead to the model becoming biased to the classes with the highest occurrences. Methods of dealing with class imbalance have become its own area of study within ML disciplines, thus, implications of solutions to such problems are mostly open-ended. Common methods of dealing with imbalance include undersampling majority classes and oversampling minority classes both of which come with their own problems. Undersampling can lead to overfitting as samples are duplicated, or synthetically generated, and oversampling can lead to removing lots of important information.

In this thesis, a combined approach including both under- and oversampling was used to balance the dataset. The results seem to indicate that the model did not overfit and still reach a high accuracy, thus, it did not remove too many important samples, however, cross-validation and other evaluation methods can be inefficient in some cases of oversampled datasets. Santos et al. 2018 suggest that model evaluation can be over-optimistic if the training and testing datasets contain much of the same data. This is common when severe oversampling of minority classes has been used. In this thesis, oversampling was used very sparingly and only in combination with additional undersampling techniques. The resulting dataset did not increase much in size, thus an almost equal amount of majority classes were removed as minority classes were synthetically generated using Synthetic Minority Oversampling Technique (SMOTE). However, further analysis into the data preparation stage and evaluation process might be warranted in order to determine the implications of this sampling process as well as further research into which sampling techniques are the most appropriate for the training dataset.

6.3.4 External impact factors

Lastly, the predictability of vessels ultimately depends on the model's ability to find global voyage patterns for different types of shipping vessels. Thus, the level of predictability can be affected by external factors that have a significant impact on these patterns. For instance, shipping traffic is orthogonal to the demand of cargo freight which reflects the production of goods and services, thus, fluctuations in production, as well as consumerism, results in fluctuations in shipping traffic and voyage patterns. Therefore, it must be considered that changes in commercial supply and demand have an effect on the validity of the presented prediction model.

Moreover, the foundation dataset used in this thesis was collected from a historical set of AIS data ranging from December 2019 to March 2021, therefore, and significant impact factor could be that of the outbreak of the *COVID-19* virus that affected the entire world in the year 2020 (Velavan and Meyer 2020). The outbreak has had a significant impact on the shipping industry in the time range of available AIS data used in the thesis. For instance, the virus outbreak has to lead to various port closures, less demand for cargo, and extensive layups (vessels brought out of commission)[1]. Oil prices have also been affected, and some countries like Norway have had an overflow of resources that could not be shipped to other countries. In these cases, tanker vessels have been recommissioned for oil storage purposes. Furthermore, a study presented Michail and Melas 2020 claims there has been a significant impact on especially the tanker and dry bulk cargo segments and found a measurable correlation between the increase in *COVID-19* cases and decrease in dry bulk and tanker freight indices.

Because of the extensive impact of the outbreak, it must be considered that the model trained for arrival port prediction presented in this thesis is also affected. In the predefined dataset, there is not enough data to measure the impact since it should at least cover a full year of data before the virus broke out for comparisons. However, given more historical AIS data ranging back further in time, a study could be conducted to investigate this relationship.

6.4 Commercial applicability

This thesis has been developed in collaboration with the maritime technological startup company Maritime Optima AS (MO) where the author is also employed and has been involved with since it was founded in 2018. Therefore, there are interests in regard to future use and commercial applicability. After the proposed model was trained and evaluated, other external shipping contacts provided by MO were also interviewed in order to gain insight into possible future applications and validity. It was clear from this information gathered that the most promising aspect of the thesis involves the possible applications toward forecasting vessel availability within specific segments in specific ports and regions.

A path toward implementing such as system would involve combining resulting arrival port predictions with a method of estimating the Estimated Time of Arrival (ETA) to the predicted arrival port from the vessel's last known position. This could relatively easily be implemented by using a route estimating tool such as the one provided by MO. This tool finds the most optimal route from any two given points at sea returning the distance of the calculated route. Thus, given a prediction of the next arrival port and ETA for every currently traveling vessel, an estimate of which vessels will be arriving at different ports at different times can be calculated. This functionality enables shipping operators and investors to make more informed decisions when deciding what cargoes to bid on and what areas to focus on. It also helps the cargo owners decide when to ship this cargo.

Today, such information is currently provided as a service through shipping brokers who conduct extensive analysis and sell their analysis to charterers, ship owners, investors, and cargo owners. A point made from one of the interviewees was that the work presented in this thesis would probably be of high

[1] `https://www.mondaq.com/marine-shipping/958770`

value to the brokers as additional insight into voyage patterns rather than to the charterers themselves. The brokers are trusted sources of information while technological solutions are less so and can currently not cover all of the services that brokers provide. Thus, the presented solution can help brokers collect data more effectively, or from a different perspective, as well as be integrated into existing software solutions such as MO.

6.5 Conclusions and future work

The shipping industry is a vast and complex ecosystem that has extensive influence on every country in the world as well as the global economy. For the companies and investors involved, being able to make effective market predictions is key to making good decisions and beneficial investments. The shipping market is volatile and mainly affected by the supply of vessels and the demand for cargo. This thesis has presented a method for vessel destination forecasting based on historical AIS which is capable of considering additional information such as vessel segments when making predictions. When applied to a global set of vessels, it could be applied to forecasting the availability of vessels in ports and regions thus providing insight into the vessel supply aspect of the shipping market. This possible application of the thesis was confirmed to be of interest to shipping experts who were interviewed to confirm the validity of the thesis.

Effectively predicting future destination ports and vessel availability is a complex problem, and there are many factors that affect vessels' voyage patterns. Therefore, the proposed method was focused on supporting multiple vessel, or voyage, features. By abstracting spatial trajectories into categorical and numerical values, additional features can be later added to investigate what information impacts the model's performance. The trajectory similarity measurement used in this abstraction can also be exchanged with another so different similarity measurements can be explored and improved upon without changing the underlying data structure.

The trajectory similarity measurement initially used in this thesis was the Symmetric Segment-Path Distance (SSPD) algorithm. When applying this algorithm to a range of historical incomplete voyages, it achieved an accuracy level of *33%*. Since the structure of the dataset provides high flexibility in replacing the SSPD-based MSTD value with another, for future work, it is suggested that different trajectory similarity measurements are implemented and evaluated. For instance, a promising method for trajectory similarity measurement was presented in Zhang et al. 2020 which could be combined with the proposed solution to improve the performance of the model.

The final trained Machine Learning (ML) model had a measured accuracy of *72%* and was applicable toward analyzing the predictability of different vessel segments and sub-segments as well as determining correlating relationships between size, capacity, and predictability. However, the evaluation process presented in the thesis could be further expanded upon to gain further insights into the validity of the model. For instance, applying a distance-based error metric to the sampled predictions could be telling as to how close the model was when predicting incorrect arrival ports.

Moreover, the thesis used a specific voyage definition when constructing voyages from the historical AIS data. The advantages of this definition have been discussed, and alternate definitions have been suggested. As a candidate for future work, it is suggested that the alternate voyage definition that takes advantage of additional port data should be explored. A combined approach using positional clustering and port data could be used to determine the intent, or context, of vessels stopping at different ports. Implications on prediction performance as well as validity for commercial users should be described during such as study.

The problem of multi-class classification on imbalanced datasets should also further be investigated to determine the most appropriate method of managing imbalance within the constructed training dataset. Santos et al. 2018 found promising results using a specific combination of SMOTE oversampling and Tomek Links undersampling to balance the dataset before training while avoiding overfitting and over-optimism in evaluation. The impact of the current sampling approach could also further be investigated as it relates to the validity of the trained model. Another approach to mitigate this issue could be to train specific models for different vessel segments. For vessels within the same segment, there are fewer possible arrival ports thus reducing the cardinality of the training datasets. This could also be beneficial to gain further insights into the predictability of different vessel segments.

Furthermore, as already mentioned, the structure of the training process and dataset provides a foundation that can easily be extended with additional features. The trained Extreme Gradient Boosting (XGBoost) model can also estimate feature importances which makes it easy to add and evaluate new features. For future work, additional features should be applied to the training set such as seasonality, or time of year, whether the traveling vessel is in a ballast (unloaded) or laden condition (loaded), and current draft (depth in water). The aforementioned features are thought to provide more insight into voyage patterns by experts interviewed as part of this thesis.

Lastly, in terms of future commercial applications, it is suggested that predictions from the presented model can be combined with a route estimator or a distance table in order to calculate the Estimated Time of Arrival (ETA) to predicted arrival ports from vessels' last known positions. Given an overview of every available vessel's predicted next arrival port and its ETA, it is possible to estimate what vessels are thought to be available at different ports and regions at different times. Thus, an overview can be produced that includes how many vessels of different segments and sub-segments are thought to arrive at different ports which has been confirmed to be of high commercial value.

6.6 Concluding remarks

This thesis has set out to investigate the topic of AIS-based vessel destination predictions and maritime logistics as it can benefit the maritime industry. Although it has its limitations, it has, hopefully, provided insights into the challenge and complexity of this topic area and shaped a foundation that can be further extended upon in both an academic and commercial sense.

Bibliography

Alizadeh, Danial, Ali Asghar Alesheikh, and Mohammad Sharif (2020). "Prediction of vessels locations and maritime traffic using similarity measurement of trajectory." In: *Annals of GIS*. ISSN: 19475691. DOI: 10.1080/19475683.2020.1840434.

Alizadeh, Danial, Ali Asghar Alesheikh, and Mohammad Sharif (2021). *Vessel Trajectory Prediction Using Historical Automatic Identification System Data*. DOI: 10.1017/S0373463320000442.

Bachar, Moti, Gal Elimelech, Itai Gat, Gil Sobol, Nicolo Rivetti, and Avigdor Gal (2018). "Grand challenge: Venilia, on-line learning and prediction of vessel destination." In: *DEBS 2018 - Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems*. DOI: 10.1145/3210284.3220505.

Besse, Philippe, Brendan Guillouet, Jean-Michel Loubes, and Royer François (2015). *Review and Perspective for Distance Based Trajectory Clustering*. arXiv: 1508.04904 [stat.ML].

Borkowski, Piotr (2017). "The ship movement trajectory prediction algorithm using navigational data fusion." In: *Sensors (Switzerland)* 17.6. ISSN: 14248220. DOI: 10.3390/s17061432.

Braca, Paolo, Enrica d'Afflisio, Leonardo Maria Millefiori, and Peter K. Willett (2018). "Detecting Anomalous Deviations from Standard Maritime Routes Using the Ornstein-Uhlenbeck Process." In: *IEEE Transactions on Signal Processing*. ISSN: 1053587X. DOI: 10.1109/TSP.2018.2875887.

Brandt, Tobias and Marco Grawunder (2017). "Moving object stream processing with short-time prediction." In: *Proceedings of the 8th ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS 2017*. DOI: 10.1145/3148160.3148168.

Brummelen, Glen Van (2013). *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry*. Princeton University Press. ISBN: 9780691148922. URL: http://www.jstor.org/stable/j.ctt1r2fvb.

Burger, Christiaan Neil, Trienko Lups Grobler, and Waldo Kleynhans (2020). "Discrete Kalman Filter and Linear Regression Comparison for Vessel Coordinate Prediction." In: *Proceedings - IEEE International Conference on Mobile Data Management*. Vol. 2020-June. DOI: 10.1109/MDM48529.2020.00062.

Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer (2002). "SMOTE: Synthetic minority over-sampling technique." In: *Journal of Artificial Intelligence Research* 16. ISSN: 10769757. DOI: `10.1613/jair.953`.

Chen, Chih-Wei, Charles Harrison, and Hsin-Hsiung Huang (2020). "The Unsupervised Method of Vessel Movement Trajectory Prediction." In: *CoRR* abs/2007.13712. arXiv: `2007.13712`. URL: `https://arxiv.org/abs/2007.13712`.

Dalsnes, Biørnar R., Simen Hexeberg, Andreas L. Flåten, Bjørn Olav H. Eriksen, and Edmund F. Brekke (2018). "The Neighbor Course Distribution Method with Gaussian Mixture Models for AIS-Based Vessel Trajectory Prediction." In: *2018 21st International Conference on Information Fusion, FUSION 2018*. DOI: `10.23919/ICIF.2018.8455607`.

Dijt, Pim and Pascal Mettes (2020). "Trajectory prediction network for future anticipation of ships." In: *ICMR 2020 - Proceedings of the 2020 International Conference on Multimedia Retrieval*. DOI: `10.1145/3372278.3390676`.

Ding, Mengzhen, Wei Su, Yingjie Liu, Jiuwen Zhang, Jianrui Li, and Jinzhao Wu (2020). "A Novel Approach on Vessel Trajectory Prediction Based on Variational LSTM." In: *Proceedings of 2020 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2020*. DOI: `10.1109/ICAICA50127.2020.9182537`.

Dobrkovic, Alexander, Maria Eugenia Iacob, Jos Van Hillegersberg, Martin R.K. Mes, and Maurice Glandrup (2015). "Towards an approach for long term AIS-based prediction of vessel arrival times." In: *Logistics and Supply Chain Innovation: Bridging the Gap between Theory and Practice*. DOI: `10.1007/978-3-319-22288-2_16`.

Dobrkovic, Andrej, Maria Eugenia Iacob, and Jos van Hillegersberg (2018). "Maritime pattern extraction and route reconstruction from incomplete AIS data." In: *International Journal of Data Science and Analytics* 5.2-3. ISSN: 23644168. DOI: `10.1007/s41060-017-0092-8`.

Dobrkovic, Andrej, Maria Eugenia Iacob, and Jos Van Hillegersberg (2015). "Using machine learning for unsupervised maritime waypoint discovery from streaming ais data." In: *ACM International Conference Proceeding Series*. Vol. 21-22-Octo. DOI: `10.1145/2809563.2809573`.

El Mekkaoui, Sara, Loubna Benabbou, and Abdelaziz Berrado (2020). "Predicting ships estimated time of arrival based on ais data." In: *ACM International Conference Proceeding Series*. DOI: `10.1145/3419604.3419768`.

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, pp. 226–231.

Forti, Nicola, Leonardo M. Millefiori, Paolo Braca, and Peter Willett (2020). "Prediction oof Vessel Trajectories from AIS Data Via Sequence-To-Sequence Recurrent Neural Networks." In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. Vol. 2020-May. DOI: `10.1109/ICASSP40776.2020.9054421`.

Ghojogh, Benyamin and Mark Crowley (2019). *The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial*. arXiv: 1905.12787 [stat.ML].

Guo, Shuai, Chao Liu, Zhongwen Guo, Yuan Feng, Feng Hong, and Haiguang Huang (2018). "Trajectory prediction for ocean vessels base on K-order multivariate markov chain." In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 10874 LNCS. DOI: 10.1007/978-3-319-94268-1_12.

Hamada, Kunihiro, Noritaka Hirata, Kai Ihara, Dimas Angga Fakhri Muzhoffar, and Mohammad Danil Arifin (2021). "Development of Basic Planning Support System Using Marine Logistics Big Data and Its Application to Ship Basic Planning." In: *Lecture Notes in Civil Engineering*. Vol. 65 LNCE. DOI: 10.1007/978-981-15-4680-8_21.

Hexeberg, Simen, Andreas L. Flaten, Bjorn Olav H. Eriksen, and Edmund F. Brekke (Aug. 2017). "AIS-based vessel trajectory prediction." In: *20th International Conference on Information Fusion, Fusion 2017 - Proceedings*. Institute of Electrical and Electronics Engineers Inc. ISBN: 9780996452700. DOI: 10.23919/ICIF.2017.8009762.

Jia, Haiying, Roar Adland, and Yuchen Wang (2019). "Latin American Oil Export Destination Choice: A Machine Learning Approach." In: *IEEE International Conference on Industrial Engineering and Engineering Management*. DOI: 10.1109/IEEM44572.2019.8978548.

Jin, Jialong, Wei Zhou, and Baichen Jiang (2020). "Maritime Target Trajectory Prediction Model Based on the RNN Network." In: *Lecture Notes in Electrical Engineering*. Vol. 572 LNEE. DOI: 10.1007/978-981-15-0187-6_39.

Jung, Hyungkun, Kang Woo Lee, and Eun Sun Cho (2019). "Outlier detection for ship trajectory prediction." In: *MobiSys 2019 - Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. DOI: 10.1145/3307334.3328639.

Karataş, Gözde Boztepe, Pinar Karagoz, and Orhan Ayran (2020). "Trajectory Prediction for Maritime Vessels Using AIS Data." In: *Proceedings of the 12th International Conference on Management of Digital EcoSystems, MEDES 2020*. DOI: 10.1145/3415958.3433079.

Kim, Kwang Il and Keon Myung Lee (2018). "Preprocessing ship trajectory data for applying artificial neural network in harbour area." In: *Proceedings - 2017 European Conference on Electrical Engineering and Computer Science, EECS 2017*. DOI: 10.1109/EECS.2017.36.

Konstantinos, Chatzikokolakis, Dimitrios Zissis, Giannis Spiliopoulos, and Konstantinos Tserpes (2018). "Mining vessel trajectory data for patterns of search and rescue." In: *CEUR Workshop Proceedings*. Vol. 2083.

Lechtenberg, Sandra, Diego Braga, and Bernd Hellingrath (Sept. 2019). "Automatic Identification System (AIS) data based Ship-Supply Forecasting." In: DOI: 10.15480/882.2487.

Lei, Po Ruey (2020). "Mining maritime traffic conflict trajectories from a massive AIS data." In: *Knowledge and Information Systems* 62.1. ISSN: 02193116. DOI: 10.1007/s10115-019-01355-0.

Li, Jiadong, Xueqi Li, and Lijuan Yu (2018). "Ship traffic flow prediction based on AIS data mining." In: *Proceedings - 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation, YAC 2018*. DOI: 10.1109/YAC.2018.8406485.

Li, Wenkai, Chunwei Zhang, Jie Ma, and Chengfeng Jia (2019). "Long-term vessel motion predication by modeling trajectory patterns with AIS data." In: *ICTIS 2019 - 5th International Conference on Transportation Information and Safety*. DOI: 10.1109/ICTIS.2019.8883596.

Lian, Yujie, Lujing Yang, Lingfeng Lu, Jing Bo Sun, and Yu Lu (2019). "Research on Ship AIS Trajectory Estimation Based on Particle Filter Algorithm." In: *Proceedings - 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2019*. Vol. 1. DOI: 10.1109/IHMSC.2019.00077.

Liu, Jiao, Guoyou Shi, and Kaige Zhu (2019). "Vessel trajectory prediction model based on ais sensor data and adaptive chaos differential evolution support vector regression (ACDE-SVR)." In: *Applied Sciences (Switzerland)* 9.15. ISSN: 20763417. DOI: 10.3390/app9152983.

Liu, Xinglong, Wei He, Jinguang Xie, and Xiumin Chu (2020). "Predicting the Trajectories of Vessels Using Machine Learning." In: *2020 5th International Conference on Control, Robotics and Cybernetics, CRC 2020*. DOI: 10.1109/CRC51253.2020.9253496.

Ma, Shexiang, Shanshan Liu, and Xin Meng (2020). "Optimized BP neural network algorithm for predicting ship trajectory." In: *Proceedings of 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2020*. DOI: 10.1109/ITNEC48623.2020.9085154.

Magdy, Nehal, Mahmoud Sakr, Tamer Abdelkader, and Khaled Elbahnasy (Dec. 2015). *Review on trajectory similarity measures*. DOI: 10.1109/IntelCIS.2015.7397286.

Mao, Shangbo, Enmei Tu, Guanghao Zhang, Lily Rachmawati, Eshan Rajabally, and Guang-Bin Huang (2018). "An Automatic Identification System (AIS) Database for Maritime Trajectory Prediction and Data Mining." In: DOI: 10.1007/978-3-319-57421-9_20.

Mestl, Thomas, Dnv Gl, Høvik Norway, and Kay Dausendschön (May 2016). "Port ETA Prediction based on AIS Data." In: *15th International Conference on Computer and IT Applications in the Maritime Industries COMPIT16*, pp. 198–201.

Michail, Nektarios A. and Kostis D. Melas (2020). "Shipping markets in turmoil: An analysis of the Covid-19 outbreak and its implications." In: *Transportation Research Interdisciplinary Perspectives* 7, p. 100178. ISSN: 2590-1982. DOI: 10.1016/j.trip.2020.100178.

Murray, Brian and Lokukaluge P. Perera (2019). "An ais-based multiple trajectory prediction approach for collision avoidance in future vessels." In: *Proceedings* *of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*. Vol. 7B-2019. DOI: 10.1115/OMAE2019-95963.

Murray, Brian and Lokukaluge P. Perera (2020). “Unsupervised trajectory anomaly detection for situation awareness in maritime navigation.” In: *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*. Vol. 6A-2020. DOI: 10.1115/OMAE2020-18281.

Murray, Brian and Lokukaluge Prasad Perera (2018). “A data-driven approach to vessel trajectory prediction for safe autonomous ship operations.” In: *2018 13th International Conference on Digital Information Management, ICDIM 2018*. DOI: 10.1109/ICDIM.2018.8847003.

Murray, Brian and Lokukaluge Prasad Perera (2020). “A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data.” In: *Ocean Engineering* 209. ISSN: 00298018. DOI: 10.1016/j.oceaneng.2020.107478.

Nguyen, Duc Duy, Chan Le Van, and Muhammad Intizar Ali (2018). “Grand challenge: Vessel destination and arrival time prediction with sequence-to-sequence models over spatial grid.” In: *DEBS 2018 - Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems*. DOI: 10.1145/3210284.3220507.

Pallotta, Giuliana, Michele Vespe, and Karna Bryan (2013). “Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction.” In: *Entropy* 15.6, pp. 2218–2245. ISSN: 1099-4300. DOI: 10.3390/e15062218.

Patmanidis, Spyridon, Iasonas Voulgaris, Elena Sarri, George Papavassilopoulos, and George Papavasileiou (2016). “Maritime surveillance, vessel route estimation and alerts using AIS data.” In: *24th Mediterranean Conference on Control and Automation, MED 2016*. DOI: 10.1109/MED.2016.7535966.

Prochazka, V. and R. Adland (2020). “Feature engineering for supply analysis in ocean transportation.” In: *IEEE International Conference on Industrial Engineering and Engineering Management*. Vol. 2020-Decem. DOI: 10.1109/IEEM45057.2020.9309749.

Rong, H., A. P. Teixeira, and C. Guedes Soares (2019). “Ship trajectory uncertainty prediction based on a Gaussian Process model.” In: *Ocean Engineering* 182. ISSN: 00298018. DOI: 10.1016/j.oceaneng.2019.04.024.

Rong, H., A. P. Teixeira, and C. Guedes Soares (2020). “Collision probability assessment based on uncertainty prediction of ship trajectories.” In: *Developments in the Collision and Grounding of Ships and Offshore Structures - Proceedings of the 8th International Conference on Collision and Grounding of Ships and Offshore Structures, ICCGS 2019*. DOI: 10.1201/9781003002420-36.

Roşca, Valentin, Paul Diac, Emanuel Onica, and Ciprian Amariei (2018). “Grand challenge: Predicting destinations by nearest neighbor search on training vessel routes.” In: *DEBS 2018 - Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems*. DOI: 10.1145/3210284.3220509.

Santos, Miriam Seoane, Jastin Pompeu Soares, Pedro Henrigues Abreu, Helder Araujo, and Joao Santos (2018). “Cross-Validation for Imbalanced Datasets:

Avoiding Overoptimistic and Overfitting Approaches [Research Frontier].” In: *IEEE Computational Intelligence Magazine* 13.4, pp. 59–76. DOI: 10.1109/MCI.2018.2866730.

Shen, K. Y., Y. J. Chu, S. J. Chang, and S. M. Chang (2020). “A study of correlation between fishing activity and AIS data by deep learning.” In: *TransNav* 14.3. ISSN: 20836481. DOI: 10.12716/1001.14.03.01.

Stopford, Martin (2008). *Maritime Economics 3e*. DOI: http://dx.doi.org/10.4324/9780203891742.

Suo, Yongfeng, Wenke Chen, Christophe Claramunt, and Shenhua Yang (2020). “A ship trajectory prediction framework based on a recurrent neural network.” In: *Sensors (Switzerland)* 20.18. ISSN: 14248220. DOI: 10.3390/s20185133.

Tafa, Lisa Natswi, Xin Su, Jiman Hong, and Chang Choi (2019). “Automatic Maritime Traffic Synthetic Route: A Framework for Route Prediction.” In: *Communications in Computer and Information Science*. Vol. 1080 CCIS. DOI: 10.1007/978-3-030-30143-9_1.

Tang, Huang, Liqiao Wei, Yong Yin, Helong Shen, and Yinghong Qi (2020). “Detection of Abnormal Vessel Behaviour Based on Probabilistic Directed Graph Model.” In: *Journal of Navigation* 73.5. ISSN: 14697785. DOI: 10.1017/S0373463320000144.

Tang, Huang, Yong Yin, and Helong Shen (2019). “A model for vessel trajectory prediction based on long short-term memory neural network.” In: *Journal of Marine Engineering and Technology*. ISSN: 20568487. DOI: 10.1080/20464177.2019.1665258.

Tomek, Ivan (1976). “Two Modifications of CNN.” English. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-6.11, pp. 769–772. DOI: 10.1109/TSMC.1976.4309452.

Tsaini, Penolope (2011). *International Shipping and World Trade*. URL: https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/4680/Tsaini.pdf.

Uney, Murat, Leonardo M. Millefiori, and Paolo Braca (2019). “Data Driven Vessel Trajectory Forecasting Using Stochastic Generative Models.” In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. Vol. 2019-May. DOI: 10.1109/ICASSP.2019.8683444.

Velavan, Thirumalaisamy P. and Christian G. Meyer (2020). “The COVID-19 epidemic.” In: *Tropical Medicine & International Health* 25.3, pp. 278–280. DOI: 10.1111/tmi.13383.

Virjonen, Petra, Paavo Nevalainen, Tapio Pahikkala, and Jukka Heikkonen (2018). “Ship Movement Prediction Using k-NN Method.” In: *Proceedings - 2018 Baltic Geodetic Congress, BGC-Geomatics 2018*. DOI: 10.1109/BGC-Geomatics.2018.00064.

Wan, Zheng, Zhu Mo, Shun Chen, and Daniel Sperling (Apr. 2016). *Pollution: Three steps to a green shipping industry : Nature News & Comment*. URL: https://www.nature.com/news/pollution-three-steps-to-a-green-shipping-industry-1.19369.

Wang, Chang, Hongxiang Ren, and Haijiang Li (2020). “Vessel trajectory prediction based on AIS data and bidirectional GRU.” In: *Proceedings - 2020 Interna-*

tional Conference on Computer Vision, Image and Deep Learning, CVIDL 2020. DOI: 10.1109/CVIDL51233.2020.00-89.

Wang, Taizheng, Chunyang Ye, Hui Zhou, Mingwang Ou, and Bo Cheng (2021). "AIS Ship Trajectory Clustering Based on Convolutional Auto-encoder." In: *Advances in Intelligent Systems and Computing*. Vol. 1251 AISC. DOI: 10.1007/978-3-030-55187-2_39.

Watawana, Thisara and Amitha Caldera (2018). "Analyse Near Collision Situations of Ships Using Automatic Identification System Dataset." In: *5th International Conference on Soft Computing and Machine Intelligence, ISCMI 2018*. DOI: 10.1109/ISCMI.2018.8703228.

Wen, Yuanqiao, Zhongyi Sui, Chunhui Zhou, Changshi Xiao, Qianqian Chen, Dong Han, and Yimeng Zhang (2020). "Automatic ship route design between two ports: A data-driven method." In: *Applied Ocean Research* 96. ISSN: 01411187. DOI: 10.1016/j.apor.2019.102049.

Xiao, Zhe, Xiuju Fu, Liye Zhang, Wanbing Zhang, Ryan Wen Liu, Zhao Liu, and Rick Siow Mong Goh (2020). "Big Data Driven Vessel Trajectory and Navigating State Prediction With Adaptive Learning, Motion Modeling and Particle Filtering Techniques." In: *IEEE Transactions on Intelligent Transportation Systems*. ISSN: 15580016. DOI: 10.1109/TITS.2020.3040268.

You, Lan, Siyu Xiao, Qingxi Peng, Christophe Claramunt, Xuewei Han, Zhengyi Guan, and Jiahe Zhang (2020). "ST-Seq2Seq: A Spatio-Temporal Feature-Optimized Seq2Seq Model for Short-Term Vessel Trajectory Prediction." In: *IEEE Access* 8. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3041762.

Zhang, Chengkai, Junchi Bin, Wells Wang, Xiang Peng, Rui Wang, Richard Halldearn, and Zheng Liu (2020). "AIS data driven general vessel destination prediction: A random forest based approach." In: *Transportation Research Part C: Emerging Technologies* 118, p. 102729. ISSN: 0968090X. DOI: 10.1016/j.trc.2020.102729.

Zheng, Changmeng, Qi Peng, and Xuemiao Xu (2020). "Heterogenous multi-source fusion for ship trajectory complement and prediction with sequence modeling." In: *Proceedings - 2020 IEEE 5th International Conference on Data Science in Cyberspace, DSC 2020*. DOI: 10.1109/DSC50466.2020.00011.

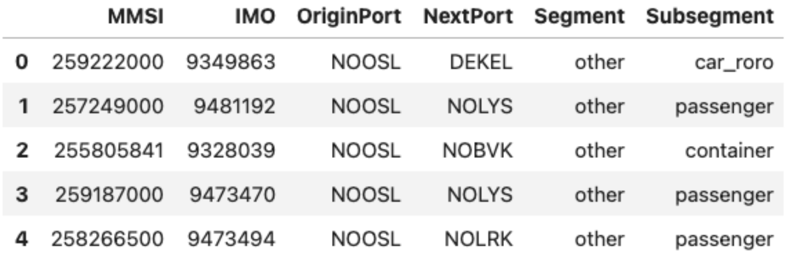
Zhou, Hai, Yaojie Chen, and Sumin Zhang (2019). "Ship Trajectory Prediction Based on BP Neural Network." In: *Journal on Artificial Intelligence* 1.1. ISSN: 2579-003X. DOI: 10.32604/jai.2019.05939.

Appendix A

Feasibility study - Summary

As already mentioned, the main motivation behind the thesis is derived from the observation that the existing methods of vessel destination prediction neglect data depth in their models. Especially, not considering the type and dimensions of vessels is presumed to be a major limitation of the existing literature. In order to establish this in an empirical manner, a feasibility study was conducted on the aspect of Maritime Optima AS (MO)'s novel segmentation of vessels. As part of the course work for the prior NTNU course called *"IMT4894 Advanced Project Work"*, such a feasibility study was conducted to estimate the impact of vessel segmentation on the aspect of port frequencies. Port frequencies, or patterns of port arrivals and departures should reflect the fact that different vessels of different types travel in different patterns. Thus, if it is possible to show that segmentations have a significant impact on these patterns through port frequencies, it can be concluded that it will have an impact on vessel destination predictions.

The dataset used in this feasibility study mainly consisted of vessel transitions, and port data. The dataset also includes the vessel's segment and subsegment. For a given port, every visiting vessel was assigned the attribute *NextPort* that indicated the next arrival port after departing the given port. Figure A.1 shows an example of vessels arriving at the port of Oslo (`NOOSL`).



	MMSI	IMO	OriginPort	NextPort	Segment	Subsegment
0	259222000	9349863	NOOSL	DEKEL	other	car_roro
1	257249000	9481192	NOOSL	NOLYS	other	passenger
2	255805841	9328039	NOOSL	NOBVK	other	container
3	259187000	9473470	NOOSL	NOLYS	other	passenger
4	258266500	9473494	NOOSL	NOLRK	other	passenger

Figure A.1: A sample of the dataset used in the feasibility study

In the feasibility study, there were two main steps in the analysis process. Firstly, a single-case analysis was conducted on a port known to the author to establish a more thorough overview of the traveling patterns of different vessel types and to gain an understanding of how to interpret the results. Secondly, a trend analysis was conducted on a collection of ports in order to establish a recurring pattern. In the study, a few major ports were selected combined with a few ports known to the author and experts in MO. The complete list of ports are listed in Appendix A.2.

A.1 Single-case analysis

For the single-case analysis, the port of Oslo (`NOOSL`) was selected as it is frequented by both dry bulk cargo vessels as well as several passenger vessels. It was presumed that the higher traveling frequency of the passenger vessels would heavily skew the most frequent next port for all vessels visiting `NOOSL`. Firstly, the distribution of the next frequented ports from the port was mapped as shown in Figure A.2 which shows that the port of Lysaker (`NOLYS`) is the most frequented next port by far. Lysaker port is a very small port that mostly receives passenger vessels that, as expected, would have high frequency because passenger vessels frequently travel back and forth over short distances. This also means that few passenger vessels could be responsible for almost all voyages, and predictions would be heavily skewed toward `NOLYS`.

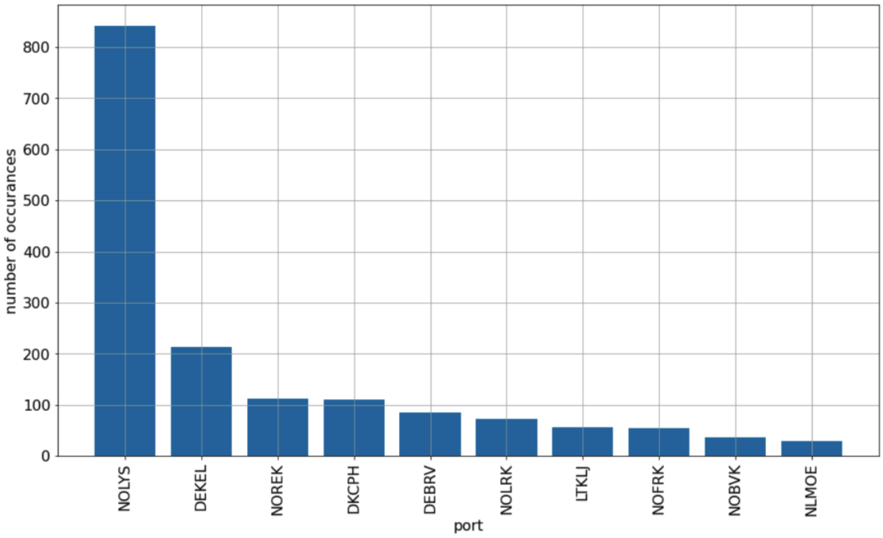
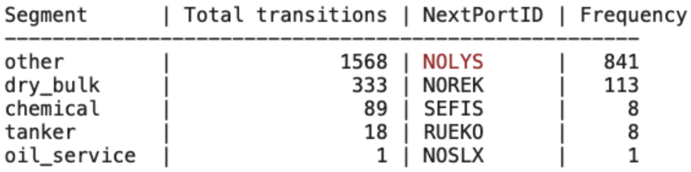


Figure A.2: Distribution of *NextPorts* from `NOOSL`

When looking into the distributions of *NextPorts* per segment it is even more apparent that the *Other* segment (which includes passenger vessels) are responsible for the high number of voyages to `NOLYS`. Figure A.3 shows this as well as the *Other* is the only segment that shares the same most frequent next port `NOLYS`. This means that a prediction algorithm using port frequencies would accurately predict the next destination ports for these other vessels, but not for the rest. Since the other vessels are responsible for 1568 out of 2009 transitions (78.05%), considering vessel segmentation for predictions, and assuming every vessel always travel to its segment most frequent next port, a prediction algorithm could also become accurate for the remainder of the vessel segments which adds up to 21.95% of all transition and probably most of the unique vessels. This is the basis used to estimate an improvement, or impact, factor for vessel segmentation on destination predictions.



```
Segment      | Total transitions | NextPortID | Frequency
-------------------------------------------------------
other        |              1568 | NOLYS      |   841
dry_bulk     |               333 | NOREK      |   113
chemical     |                89 | SEFIS      |     8
tanker       |                18 | RUEKO      |     8
oil_service  |                 1 | NOSLX      |     1
```

Figure A.3: Distribution of *NextPorts* from NOOSL per segment

Furthermore, as Figure A.4 shows, when looking at the port frequency of the dry bulk cargo vessels, it is apparent that NOLYS is not even a contender for the most frequent next port. Therefore, a prediction method considering port frequencies would not be able to accurately predict the next destination port for any other vessel other than passenger vessels.

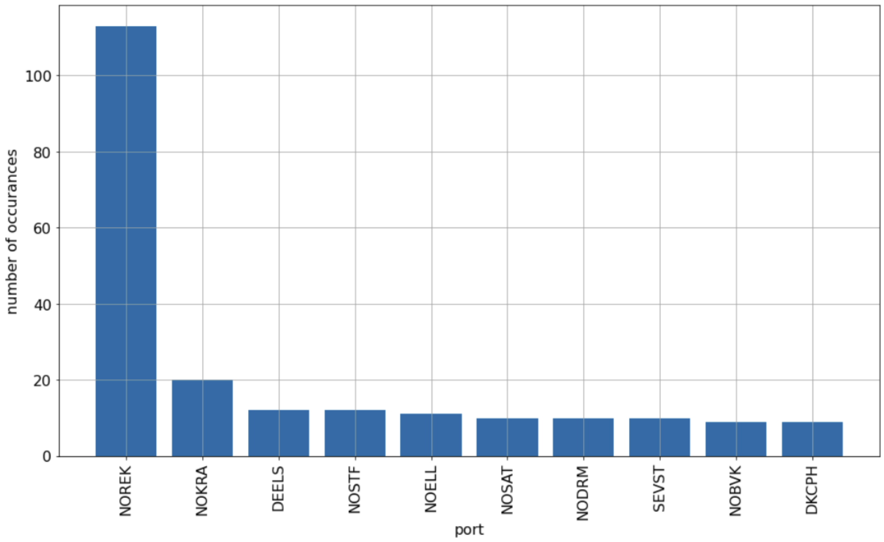
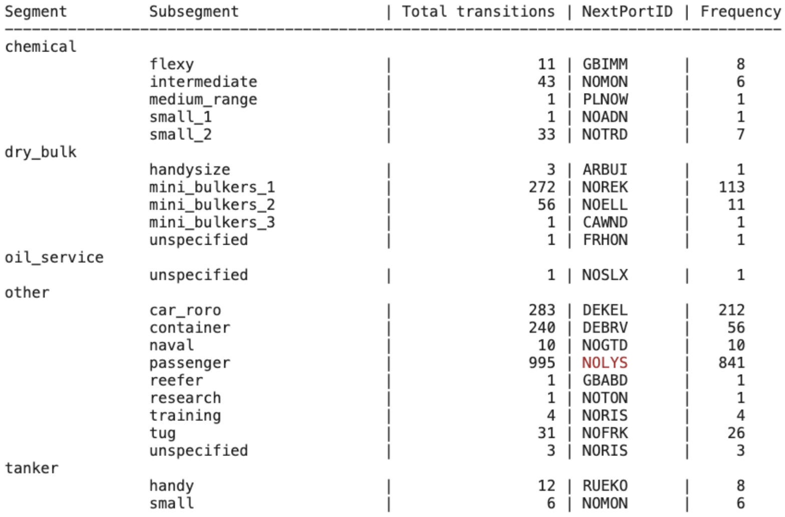


Figure A.4: Distribution of *NextPorts* from NOOSL for the *dry bulk* segment

Investigating sub-segments further confirms that a few numbers of vessels are responsible for most of all total transitions. Figure A.5 shows that the specific sub-segment *'other - passenger'*, or passenger vessels, are responsible for 49.52% of all transitions and nearly all voyages arrive at NOLYS after NOOSL. This means a prediction model could potentially be improved by 50% if it would be aware of the sub-segment of each vessel for this particular port. NOOSL seems to be a port that shows the problem area quite well because it is a smaller port that receives a lower number of different vessels, and when there are multiple passenger vessels frequently arriving at it, they heavily skew the results in their favor.



```
Segment        Subsegment                | Total transitions | NextPortID | Frequency
-------------------------------------------------------------------------------------
chemical
               flexy                     |                11 | GBIMM      |     8
               intermediate              |                43 | NOMON      |     6
               medium_range              |                 1 | PLNOW      |     1
               small_1                   |                 1 | NOADN      |     1
               small_2                   |                33 | NOTRD      |     7
dry_bulk
               handysize                 |                 3 | ARBUI      |     1
               mini_bulkers_1            |               272 | NOREK      |   113
               mini_bulkers_2            |                56 | NOELL      |    11
               mini_bulkers_3            |                 1 | CAWND      |     1
               unspecified               |                 1 | FRHON      |     1
oil_service
               unspecified               |                 1 | NOSLX      |     1
other
               car_roro                  |               283 | DEKEL      |   212
               container                 |               240 | DEBRV      |    56
               naval                     |                10 | NOGTD      |    10
               passenger                 |               995 | NOLYS      |   841
               reefer                    |                 1 | GBABD      |     1
               research                  |                 1 | NOTON      |     1
               training                  |                 4 | NORIS      |     4
               tug                       |                31 | NOFRK      |    26
               unspecified               |                 3 | NORIS      |     3
tanker
               handy                     |                12 | RUEKO      |     8
               small                     |                 6 | NOMON      |     6
```

Figure A.5: Distribution of *NextPorts* from `NOOSL` per sub-segment

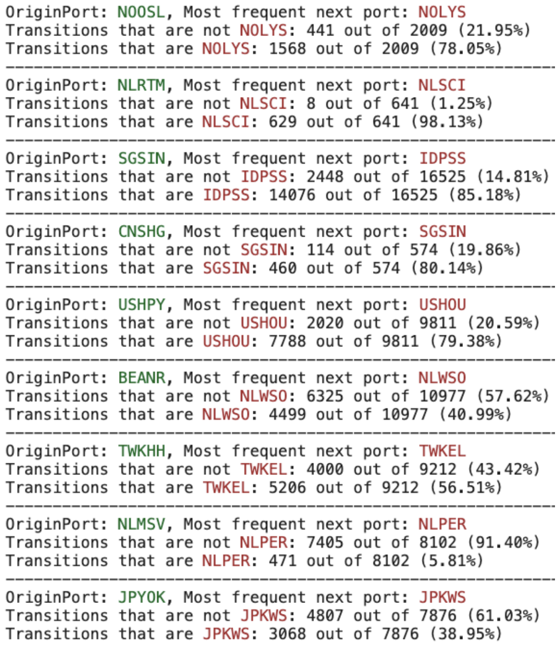
A.2 Trend analysis

As already mentioned, for the trend analysis, a number of ports were selected based on their size and traffic. There were also a couple of known ports included in this dataset to easier interpret the results. The ports used in the analysis were:

- `NLRTM` — Rotterdam, Netherlands
- `NOOSL` — Oslo, Norway
- `CNSHG` — Shanghai, China
- `NLMSV` — Maasvlakte, Netherlands
- `SGSIN` — Singapore, Singapore
- `USHPY` — Baytown, USA
- `BEANR` — Antwerpen, Belgium
- `TWKHH` — Kaohsiung, Taiwan
- `JPYOK` — Yokohama, Japan

The same process as for the single-case analysis was conducted, but on a higher level as the main purpose of this study was to establish a trend in terms of a impact factor of vessel segmentation on port frequencies. Figure A.6 shows a similar version of the table used for the single port analysis (Figure A.2) but also shows the number of transitions that differed from the most frequent next port when considering segments (i.e. the estimated improvement factor).

It is apparent that there are variances in improvement factors for dif-



```
OriginPort: NOOSL, Most frequent next port: NOLYS
Transitions that are not NOLYS: 441 out of 2009 (21.95%)
Transitions that are NOLYS: 1568 out of 2009 (78.05%)
-----------------------------------------------------------
OriginPort: NLRTM, Most frequent next port: NLSCI
Transitions that are not NLSCI: 8 out of 641 (1.25%)
Transitions that are NLSCI: 629 out of 641 (98.13%)
-----------------------------------------------------------
OriginPort: SGSIN, Most frequent next port: IDPSS
Transitions that are not IDPSS: 2448 out of 16525 (14.81%)
Transitions that are IDPSS: 14076 out of 16525 (85.18%)
-----------------------------------------------------------
OriginPort: CNSHG, Most frequent next port: SGSIN
Transitions that are not SGSIN: 114 out of 574 (19.86%)
Transitions that are SGSIN: 460 out of 574 (80.14%)
-----------------------------------------------------------
OriginPort: USHPY, Most frequent next port: USHOU
Transitions that are not USHOU: 2020 out of 9811 (20.59%)
Transitions that are USHOU: 7788 out of 9811 (79.38%)
-----------------------------------------------------------
OriginPort: BEANR, Most frequent next port: NLWSO
Transitions that are not NLWSO: 6325 out of 10977 (57.62%)
Transitions that are NLWSO: 4499 out of 10977 (40.99%)
-----------------------------------------------------------
OriginPort: TWKHH, Most frequent next port: TWKEL
Transitions that are not TWKEL: 4000 out of 9212 (43.42%)
Transitions that are TWKEL: 5206 out of 9212 (56.51%)
-----------------------------------------------------------
OriginPort: NLMSV, Most frequent next port: NLPER
Transitions that are not NLPER: 7405 out of 8102 (91.40%)
Transitions that are NLPER: 471 out of 8102 (5.81%)
-----------------------------------------------------------
OriginPort: JPYOK, Most frequent next port: JPKWS
Transitions that are not JPKWS: 4807 out of 7876 (61.03%)
Transitions that are JPKWS: 3068 out of 7876 (38.95%)
```

Figure A.6: Port frequencies and transition distribution as they relate to the most frequent next port for the selected ports

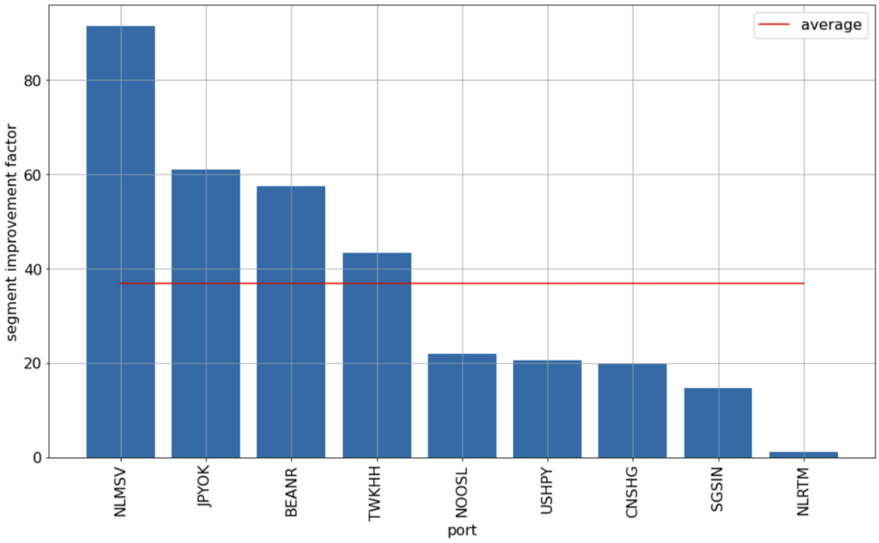


Figure A.7: Distribution of improvement factors for each origin port considering segments

ferent ports ranging from as low as *1.25%* to as high as *91.40%*. In the case of NLRTM, which is mostly a dry bulk port, there were no considerable improvements as almost all vessels are of the same segment. For the port NLMSV, the opposite was the case as there were a plethora of different types of vessels that frequented the port. Figure A.7 shows the distribution of the improvement factor considering segments for each origin port as well as the overall average impact factor for these 9 ports which was *36.88%*.

Furthermore, when looking at the impact of sub-segments, as Figure A.8 shows, it seems that the improvement factor has increased overall. For example, in the case of NLRTM, the improvement factor has increased from *1.25%* to *19.66%*, and although this varied for the different ports, the overall average improvement factor increased from *36.88%* to *50.28%*.

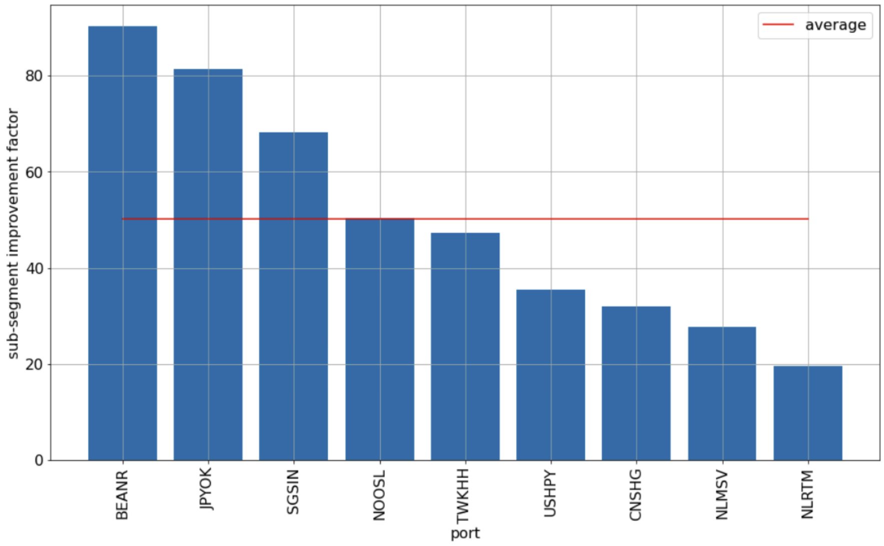


Figure A.8: Distribution of improvement factors for each origin port considering sub-segments

A prediction method considering the frequencies of ports for vessel destination predictions would choose the most frequent next port for the predicted next destination. In this scenario, ignoring the vessel's type (segmentation) would give the wrong prediction for a lot of vessels from different segments in a lot of ports. The results from the feasibility study clearly indicates that applying the aspect of vessel segmentation to such models would definitively have an impact on prediction accuracy and, therefore, is worth investigating further.

Appendix B

Trajectory sampler Golang package

Code listing B.1: Trajectory sampler package

```go
// Package sampler can be used to sample 2D trajectories based on distance as well as 3D trajectories based on time
package sampler

import (
        "errors"
        "sort"
        "time"

        "github.com/paulmach/orb"
        "github.com/paulmach/orb/encoding/wkt"
        "github.com/paulmach/orb/geo"
        "github.com/paulmach/orb/resample"
        "github.com/twpayne/go-geom"
        "github.com/twpayne/go-geom/encoding/ewkbhex"
        geomwkt "github.com/twpayne/go-geom/encoding/wkt"
)

// Metric either time or distance used for sampling
type Metric string

const (
        // Time sampling uses hours as SamplRate unit
        Time Metric = "time"
        // Distance sampling uses meters as SamplRate unit
        Distance Metric = "distance"
)

// Instance of sampler
type Instance struct {
        // Trajectory in (E)WKT format, e.g. (LINESTRING Z (x, y, timestamp))
        WKTTrajectory string
        // Trajectory in (E)WBT format, e.g. (01020000800801058A1A4CC312424025B5548F949B5454240000)
        WKBHexTrajectory string
        Metric           Metric
        // SampleRate unit is hours for time and meters for distance
        SampleRate int
```

```go
}

// Parsers

// parse2DTrajectory parses a trajectory not containing Z coordinates
// we want orb.Geometry for the resampling method which doesn’t support 3D geometry,
// so we use parse3DTrajectory and flatten the coords
func (s *Instance) parse2DTrajectory() (orb.LineString, error) {
        geoLine, err := s.parse3DTrajectory()
        if err != nil {
                return nil, err
        }
        lineString := orb.LineString{}

        coords := geoLine.Coords()
        for _, c := range coords {
                lineString = append(lineString, orb.Point{c[0], c[1]})
        }

        return lineString, nil
}

// parse3DTrajectory parses a 3D trajectory where the Z coordinate contains time
func (s *Instance) parse3DTrajectory() (*geom.LineString, error) {
        var (
                geometry geom.T
                err      error
        )

        if s.WKBHexTrajectory == "" && s.WKTTrajectory == "" {
                return nil, errors.New("specify either WKTTrajectory, or WKBHexTrajectory")
        }

        if s.WKBHexTrajectory != "" {
                geometry, err = ewkbhex.Decode(s.WKBHexTrajectory)
                if err != nil {
                        return nil, err
                }
        }

        if s.WKTTrajectory != "" {
                geometry, err = geomwkt.Unmarshal(s.WKTTrajectory)
                if err != nil {
                        return nil, err
                }
        }

        line, ok := geometry.(*geom.LineString)
        if !ok {
                return nil, errors.New("geometry was not a valid linestring")
        }

        return line, nil
}

// Resample runs resampling based on given config
func (s *Instance) Resample() (string, error) {
        switch s.Metric {
        case Distance:
                return s.resampleDistance()
```

```go
	case Time:
		return s.resampleTime()
	default:
		return "", errors.New("metric was not specified to a valid metric")
	}
}

func (s *Instance) resampleDistance() (string, error) {
	parsed, err := s.parse2DTrajectory()
	if err != nil {
		return "", err
	}

	sampled := resample.ToInterval(parsed, geo.DistanceHaversine, float64(s.SampleRate))
	return wkt.MarshalString(sampled), nil
}

// resampleTime resamples trajectory based on s.SampleRate given in hours.
// Extracts the first position within intervals based on sample rate
func (s *Instance) resampleTime() (string, error) {
	var err error

	trajectory, err := s.parse3DTrajectory()
	if err != nil {
		return "", err
	}

	intervals := s.getTimeIntervals(trajectory)
	reducedCoords := []geom.Coord{}

	coords := trajectory.Coords()

	// within each interval add the first coord to reducedCoords
	for _, interval := range intervals {
		var first *geom.Coord

		// find coord first within interval
		for i := range coords {
			coordInterval := s.roundTime(int64(coords[i][2]))
			if coordInterval == interval {
				first = &coords[i]
				break
			}
		}

		if first != nil {
			reducedCoords = append(reducedCoords, *first)
		}
	}

	// if the last coord wasn’t in the reduced coords, add it
	lastReduced := reducedCoords[len(reducedCoords)-1]
	if !lastReduced.Equal(geom.XYZ, coords[len(coords)-1]) {
		reducedCoords = append(reducedCoords, coords[len(coords)-1])
	}

	if len(reducedCoords) <= 1 {
		return "", errors.New("too few points in sampled trajectory")
	}

	reduced, err := geom.NewLineString(geom.XYZ).SetCoords(reducedCoords)
	if err != nil {
		return "", err
	}

	return geomwkt.Marshal(reduced)
}

// time sampling helpers

func (s *Instance) roundTime(ts int64) time.Time {
	return time.Unix(ts, 0).UTC().Round(time.Duration(s.SampleRate) * time.Hour)
}

func (s *Instance) getTimeIntervals(trajectory *geom.LineString) []time.Time {
	times := make(map[string]time.Time)
	for _, coord := range trajectory.Coords() {
		rounded := s.roundTime(int64(coord[2]))
		times[rounded.Format("2006.01.02:15:04")] = rounded
	}

	ret := make([]time.Time, 0, len(times))
	for _, value := range times {
		ret = append(ret, value)
	}

	sort.Slice(ret, func(i, j int) bool {
		return ret[i].Before(ret[j])
	})

	return ret
```

Morten Omholt-Jensen

Vessel destination forecasting based on historical AIS data

NTNU

Kunnskap for en bedre verden

MARITIME OPTIMA